Diffusion Guided Propagation Augmentation for Popularity Prediction

Chaozhuo Li[1] , **Tianqi Yang**[1*] , **Litian Zhang**[1] , **Xi Zhang**[1†]
[1]Key Laboratory of Trustworthy Distributed Computing and Service (MoE),
Beijing University of Posts and Telecommunications, China
{lichaozhuo, yangtianqi2022, zhangx}@bupt.edu.cn, litianzhang@buaa.edu.cn

Abstract

The prediction of information popularity propagation is critical for applications such as recommendation systems, targeted advertising, and social media trend analysis. Traditional approaches primarily rely on historical cascade data, often sacrificing timeliness for prediction accuracy. These methods capture aggregate diffusion patterns but fail to account for the complex temporal dynamics of early-stage propagation. In this paper, we introduce Diffusion Guided Propagation Augmentation(DGPA), a novel framework designed to improve early-stage popularity prediction. DGPA models cascade dynamics by leveraging a generative approach, where a temporal conditional interpolator serves as a noising process and forecasting as a denoising process. By iteratively generating cascade representations through a sampling procedure, DGPA effectively incorporates the evolving time steps of diffusion, significantly enhancing prediction timeliness and accuracy. Extensive experiments on benchmark datasets from Twitter, Weibo, and APS demonstrate that DGPA outperforms state-of-the-art methods in early-stage popularity prediction.

1 Introduction

Information propagation, often termed as an information cascade, is a widespread phenomenon in online social networks that describes the dynamic process through which messages are accessed and disseminated by users. Popularity prediction aims to quantify the level of attention a message will receive, typically measured by the volume of retweets or shares [Leskovec *et al.*, 2007]. Accurately predicting the popularity of information cascades is highly valuable, as it provides deeper insights into the virality of messages or products, thereby informing more effective recommendations and targeted advertising strategies [Kempe *et al.*, 2003]

Existing popularity prediction approaches can be summarized into two categories.The first category comprises feature-based approaches [Chen *et al.*, 2022], where researchers manually extract specific features, such as content quality, publisher information,and publication time, and other features from observed cascades to predict popularity. As illustrated in Figure 1(a), these methods enable early detection of information diffusion without requiring extensive observation. However, due to the intricate design of feature selection, these methods often suffer from limited accuracy. The second category includes deep learning-based approaches, In recent years, with the advent of deep learning, researchers have developed models capable of capturing the intrinsic characteristics of information diffusion. Such as recurrent neural networks (RNNs) including LSTM and GRU [Cho *et al.*, 2014]are used to model sequential processes, while graph neural networks (GNNs) capture the underlying graph structures[Lu *et al.*, 2023; Xu *et al.*, 2021; Sun *et al.*, 2022]. As shown in Figure 1(b), these models significantly enhance predictive accuracy. However, they depend heavily on prolonged observation periods, and their performance diminishes when the observation window is shortened, posing challenges for meeting the real-time demands of information dissemination.

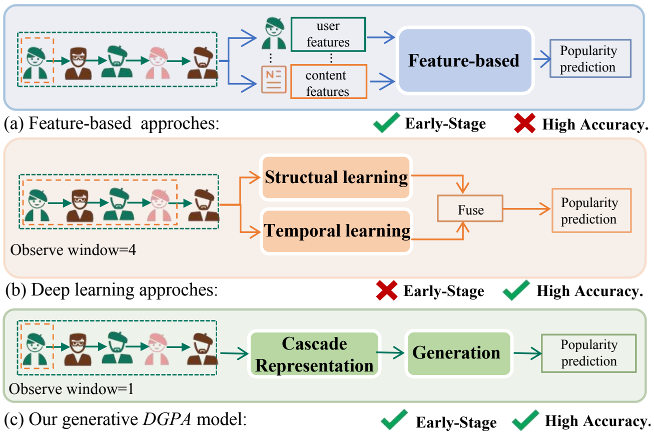


Figure 1: Comparison of popularity predition methods

This brings us a question: *Can we leverage the rich social context embedded within cascades to enhance early-stage popularity prediction?* Different from existing models,our motivation lies in generating the forthcoming cascades representation up to a specified timestamp based on the propa-

[*]Tianqi Yang is the first student author.
[†]Xi Zhang is the corresponding author.

gation patterns gleaned from previous propagation, as illustrated in Figure 1(c). our objective is to discern the underlying dynamics beneath the cascade's evolution, which enables the simulation of information diffusion, thus facilitating the early detection of emerging information. This generation-based strategy does not necessitate long-time observed cascades and is anticipated to yield rich structural and temporal propagation features.

However, designing generative models for cascade representation is non-trivial for several reasons. First, accurately modeling the distribution of discrete cascades is challenging for generative models. For instance, GAN-based approaches often face stability issues [Cao *et al.*, 2019; Lee *et al.*, 2021], while VAE-based methods suffer from posterior collapse [Tang *et al.*, 2021a; Zhao *et al.*, 2019]. Second, real-world information diffusion is irregularly sampled; the temporal sequence of user activities is non-uniform, and generating continuous time series while handling missing intermediate points remains difficult. This limitation hinders the model's ability to effectively capture dynamic changes between observations, as seen in conditional-diffusion models like CasDO[Cheng *et al.*, 2024]. Third, generative models often struggle to align generated dynamics with the actual cascade propagation processes, leading to divergence from real-world behaviors over time.

To address these challenges, we propose Diffusion Guided Propagation Augmentation for Popularity Prediction (DGPA), a novel generative framework leveraging a diffusion-based backbone. Unlike traditional approaches, DGPA integrates temporal dynamics into the diffusion process. Specifically, we employ a time-conditioned interpolation network in the forward process to bridge sparse cascade snapshots, ensuring temporally coherent intermediate representations. In the reverse process, DGPA predicts user-specific characteristics at designated timestamps. By aligning generated representations with specific time points, DGPA reduces error propagation.

Our contributions are summarized as follows:

- We propose DGPA, a novel generative model designed to generate realistic cascade representations for early-stage popularity prediction, overcoming key challenges in discrete cascade modeling.
- We develop a strategy to align generated time points with specific timestamps, reducing computational complexity and alleviating error accumulation during the generation process.
- Extensive experiments on real-world datasets demonstrate the effectiveness of DGPA, outperforming SOTA baselines in early-stage popularity prediction tasks.

2 Problem Formulation

Cascade Snapshot. A cascade characterizes the process by which a message m disseminates among a set of users $\mathcal{U}$. The initial broadcast of m by user u_0 at time t_0 is denoted as (u_0, t_0), signifying that u_0 initiates the cascade at time t_0. Subsequent transmission events, where user v_i forwards the message m received from u_i at time t_i, are represented as (u_i, v_i, t_i). The cascade's state at a particular time t_o is expressed as $c_{t_o} = \{(u_i, v_i, t_i) \mid t_i < t_o\}$.

Cascade snapshots are extracted at uniform time intervals, denoted as timesteps. For example, the i^{th} snapshot corresponds to the time $t_o = t_0 + i \cdot |timestep|$ and is denoted as c_i. If we aim to extract h snapshots, the resulting collection of cascade snapshots is represented as $C = \{(c_i, i) \mid 0 \leq i < h\}$.

Early Popularity Prediction. Building on the aforementioned definitions, we derive the set of snapshots C. In scenarios where early prediction is required, the number of available snapshots may be limited, indicating that the value of i remains small. By employing our generative DGPA approach, we can generate reasonably cascade representation after h additional timesteps, referred to as $\hat{c}_h$. This refined representation $\hat{c}_h$ allows us to effectively estimate the popularity increment ΔP up to the prediction time t_p, even when constrained by a brief observation window.

3 Model

Figure 2 illustrates the framework of the proposed **DGPA** model. The framework of the proposed DGPA model consists of three core modules:the Cascade Representation Module, which generates embedding for cascade snapshots by aggregating both the temporal and structural information of the participating users; the Dynamic Cascade Generation Module, which simulates temporal dynamics and generates cascade representations over subsequent time periods. This module employs a diffusion model as its backbone, coupling the diffusion steps within the model to the time steps in cascades [Li *et al.*, 2018]. At a high level, it treats dynamic temporal interpolation as a forward process and next-user forecasting as a denoising process. Finally, the Prediction Module utilizes the cascade representations generated by the previous modules to predict the increment in popularity [Li *et al.*, 2021].

3.1 Cascade Representation Learning Module

The Cascade Representation Learning module serves as the foundation for generating cascade embeddings by aggregating representations of participating users.This module is divided into three submodules: Temporal Learning, Structural Learning, and Embedding Fusion.

Temporal Learning

By modeling the cascade as a sequence of user engagements, this submodule is designed to extract temporal patterns that are crucial for comprehending cascade evolution trends.

For a given cascade observed within a specified observation time t_o, we sequentially organize the cascade based on the chronological order of user participation, forming a user-time sequence $\mathcal{S}_l = \{(u_i, t_i) | t_i < t_o\}$. This sequence is then input into the Transformer for encoding. The Transformer processes this input and computes the temporal representation s_{u_i} for each user u_i based on the entire sequence up to time t_i. The encoding process can be mathematically represented as follows:

$$s_{u_i} = \text{Transformer}\left(\{(u_1, t_1), (u_2, t_2), \ldots, (u_i, t_i)\}\right)_i \quad (1)$$

This results in a temporally encoded sequence $\mathcal{S} = \{(s_{u_i}, t_i) \mid t_i < t_o\}$.

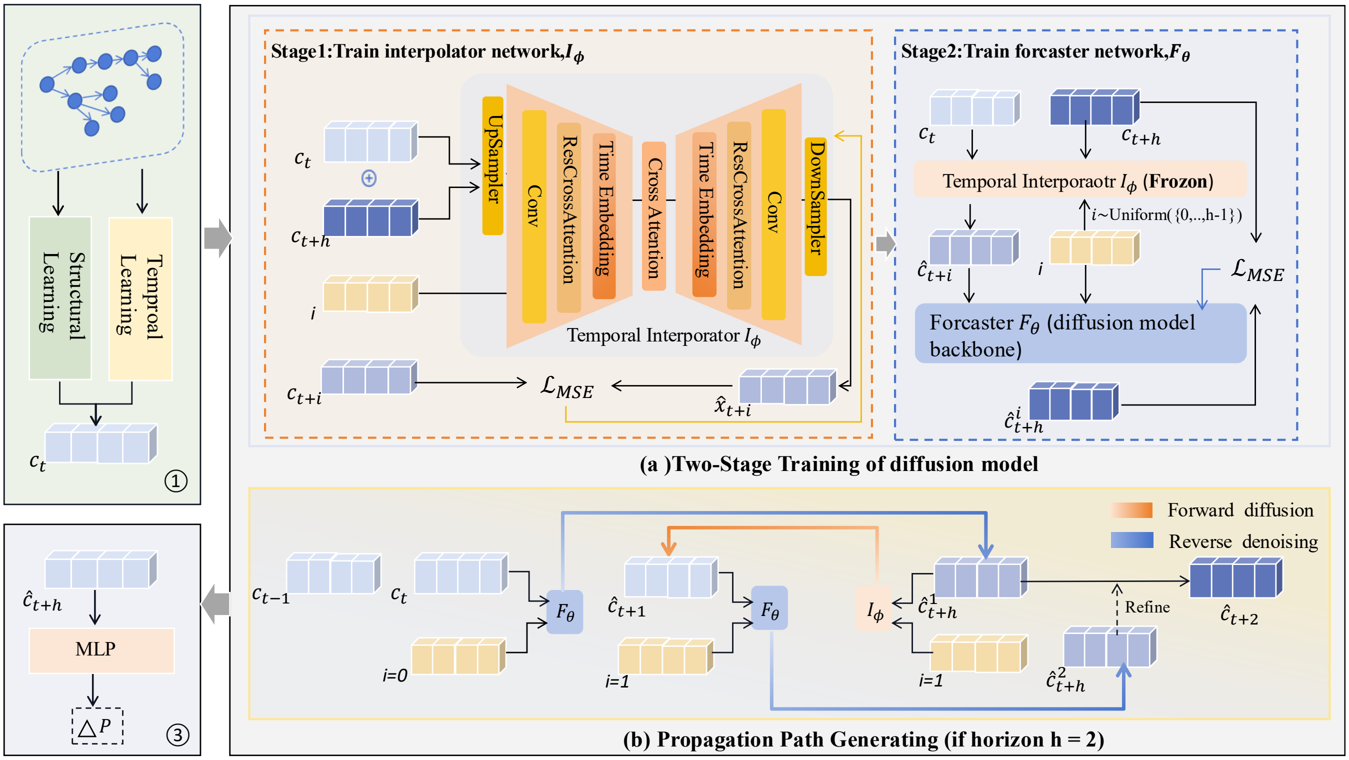


Figure 2: The overview framework of DGPA model.

Structural Learning

The Structural Learning submodule leverages a Graph Attention Network (GAT) to effectively capture the structural information inherent within the cascade graph. Specifically, given the input $\mathcal{G}_l = \{(u_i, v_i, t_i) \mid t_i < t_o\}$, the cascade can be represented as a directed acyclic graph (DAG) with a root node. This graph is then input into the GAT to update the representation of each node. The update process for each node's representation can be mathematically expressed as follows:

$$h'_{u_i} = W h_{u_i} \tag{2}$$

$$e_{ij} = \text{LeakyReLU}\left(a^T [h'_{u_i} \| h'_{v_i}]\right) \tag{3}$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}(u_i)} \exp(e_{ik})} \tag{4}$$

$$h''_{u_i} = \sigma \left(\sum_{j \in \mathcal{N}(u_i)} \alpha_{ij} h'_{v_j} \right) \tag{5}$$

$$g_{u_i} = \|_{k=1}^{K} h''_{u_i,k} \tag{6}$$

Here, g_{u_i} denotes the updated representation of node u_i after the application of the GAT. The final output is the structurally encoded graph $G^t_o = \{(g_{u_i}, t_i) \mid t_i < t_o\}$.

Embedding Fusion

The Embedding Fusion submodule combines the temporal and structural representations, S^t_o and G^t_o, by concatenating them according to the corresponding timestamps. The fusion process is mathematically represented as follows:

$$F^t = \{([s_{u_i} \oplus g_{u_i}], t_i) \mid t_i < t_o\} \tag{7}$$

Here, $\oplus$ denotes the concatenation operation.

Given a pre-defined timestep length, we select l snapshots from the sequence F^t by considering the last observation time t_o and moving backward through the sequence. Specifically, we select representations at intervals of the timestep from the sequence F^t, resulting in the set of snapshots $\{c_{t-l+1}, c_{t-l+2}, \ldots, c_t\}$. Each snapshot c_i represents the temporal embedding of the cascade at timestep i. In case of missing values, the corresponding snapshot is interpolated using the nearest neighboring values.

In the subsequent module, these l snapshots will be used to learn the conditional distribution $P(c_{t+1:t+h} \mid c_t)$ where $h < l$. This enables the model to predict h future steps while leveraging the historical context provided by the l snapshots.

3.2 Dynamic Cascade Generation Module

Inspired by the diffusion model, we integrate the temporal dynamics inherent in the data with the model's diffusion steps. The training process is divided into two distinct stages, akin to the noise addition and denoising process in diffusion models, as illustrated in Figure 2(b). First, we train a temporal interpolator for forward interpolation. In the second stage, we freeze the temporal interpolator and train a predictor to generate a cascade representation at a specified timestep. Ultimately, the trained model produces the cascade representations through sampling.

Temporal Interpolation as Forward Diffusion

The temporal interpolation process leverages a time-conditioned network $\mathcal{I}_\phi$ to interpolate between cascade snapshots. Given a cascade snapshot sequence $\mathcal{C} = \{(c_i, i) \mid 0 \leq i < l\}$, we sample a cascade representation c_t and its subsequent representation at timestep c_{t+h}. The objective is to generate intermediate cascade representations at time $t + i$, where i is a randomly chosen step within the horizon h. This is formally expressed as:

$$\mathcal{I}_\phi\left(c_t, c_{t+h}, i\right) \approx c_{t+i}, \quad i \in \{1, \dots, h-1\} \tag{8}$$

The optimization objective for the interpolation network is defined as:

$$\min_{\phi} \mathbb{E}_{i\sim\mathcal{U}[1,h-1],c_t,c_{t+i},c_{t+h}\sim\mathcal{C}}\left[\left\|\mathcal{I}_\phi\left(c_t, c_{t+h}, i\right) - c_{t+i}\right\|^2\right] \tag{9}$$

where $\mathcal{U}[1, h-1]$ represents a uniform distribution over the horizon.

The cascade representation c_{t+i} is generated by concatenating the conditioning variable i with the representations c_t and c_{t+h}. This concatenated input is processed through an encoder-decoder architecture. The encoder, consisting of convolutional layers, residual cross-attention mechanisms, and time embeddings, progressively encodes the input. The decoder, symmetric to the encoder, reconstructs the interpolated output. The interpolation stage applies nearest-neighbor interpolation to produce the final output $\hat{c}_{t+i}$. The process is represented as:

$$\hat{c}_{t+i} = \text{Interpolation}\left(\text{Decoder}\left(\text{Encoder}\left([i; c_t; c_{t+h}]\right)\right)\right) \tag{10}$$

Forecasting as Reverse Denoising

In the second training phase, the forecasting network F_θ is optimized to recover the future state c_{t+h} from partially diffused representations. This is formulated as a supervised regression task:

$$F_\theta\left(\mathcal{I}_\phi\left(c_t, c_{t+h}, i_n \mid \xi\right), i_n\right) \approx c_{t+h}, \quad i_n \in S = \{i_n\}_{n=0}^{N-1}, \tag{11}$$

Here, S denotes a temporal schedule that defines the correspondence between the diffusion steps and intermediate timesteps. The interpolation network $\mathcal{I}_\phi$ is fixed during this stage, and stochasticity is introduced via a noise variable ξ, modeled as randomly dropped weights to reflect inference uncertainty. This stochastic input is omitted from further notation for clarity.

The optimization objective for the forecaster is defined as:

$$\min_{\theta} \mathbb{E}_{n\sim\mathcal{U}[0,N-1],\, c_t,c_{t+h}\sim\mathcal{C}} \left[\left\|F_\theta\Big(\mathcal{I}_\phi(c_t, c_{t+h}, i_n \mid \xi),\ i_n\Big) - c_{t+h}\right\|^2\right] \tag{12}$$

To generalize across different levels of prediction difficulty, we include the initial state as a special case by setting $i_0 := 0$ and $\mathcal{I}_\phi\left(c_t, \cdot, i_0\right) := c_t$. This allows supervision over all timesteps in the defined temporal resolution, where typically $N = h$ and $S = \{j\}_{j=0}^{h-1}$. The schedule must satisfy $0 = i_0 < i_n < i_m < h$ for all valid $0 < n < m \leq N - 1$.

Algorithm 1 Two-stage Training

```
Input: Forecasting model $F_\theta$, Interpolator $I_\phi$, norm $\|\cdot\|$, prediction horizon $h$, timestep schedule $\{i_n\}_{n=0}^{N-1}$, weighting factors $\lambda_1$, $\lambda_2$, and auxiliary condition function $cond$
    /* Stage 1: Training the interpolator, $I_\phi$ */
Select temporal offset $i$ uniformly from $\{1, \dots, h-1\}$
Sample $\mathbf{c}_t$, $\mathbf{c}_{t+i}$, $\mathbf{c}_{t+h}$ from training set $\mathcal{C}$
Update $I_\phi$ by minimizing:
    $\mathcal{L}_\phi = \|I_\phi(\mathbf{c}_t, \mathbf{c}_{t+h}, i) - \mathbf{c}_{t+i}\|^2$
    /* Stage 2: Training the diffusion-based forecaster, $F_\theta$ */
Freeze the parameters of $I_\phi$ and introduce stochasticity during inference (e.g., dropout)
Draw a timestep index $n \in \{0, \dots, N-1\}$ and retrieve $\mathbf{c}_t$, $\mathbf{c}_{t+h}$ from dataset
Compute initial prediction:
    $\hat{\mathbf{c}}_{t+h}^{(1)} \leftarrow F_\theta\left(I_\phi(\mathbf{c}_t, \mathbf{c}_{t+h}, i_n), i_n, cond(\mathbf{c}_t, n)\right)$
If $n < N-1$, apply one-step refinement:
    $\hat{\mathbf{c}}_{t+h}^{(2)} \leftarrow F_\theta\left(I_\phi(\hat{\mathbf{c}}_{t+h}^{(1)}, \mathbf{c}_{t+h}, i_{n+1}), i_{n+1}, cond(\mathbf{c}_t, n+1)\right)$
Otherwise, set $\hat{\mathbf{c}}_{t+h}^{(2)} \leftarrow \mathbf{c}_{t+h}$
Minimize the joint objective:
    $\mathcal{L}_\theta = \lambda_1 \left\|\hat{\mathbf{c}}_{t+h}^{(1)} - \mathbf{c}_{t+h}\right\|^2 + \lambda_2 \left\|\hat{\mathbf{c}}_{t+h}^{(2)} - \mathbf{c}_{t+h}\right\|^2$
```

Given the structural resemblance between the forecaster and a denoising network in diffusion models, we refer to F_θ as the diffusion backbone. Unlike standard diffusion models that use step index n , we condition F_θ on the interpolation index i_n , allowing flexible scheduling during training and inference—even supporting timesteps unseen during training.

Because the interpolator $\mathcal{I}_\phi$ is fixed during this phase, imperfect predictions $\hat{c}_{t+h} = F_\theta\left(\mathcal{I}_\phi\left(c_t, c_{t+h}, i_n\right), i_n\right)$ can accumulate errors in sequential forecasting. To mitigate this, we introduce a one-step look-ahead loss:

$$\left\|F_\theta\left(\mathcal{I}_\phi\left(c_t, \hat{c}_{t+h}, i_{n+1}\right), i_{n+1}\right) - c_{t+h}\right\|^2, \tag{13}$$

This term, combined with the standard prediction loss, ensures temporal consistency across steps. Moreover, we optionally provide the forecaster with clean or noised versions of the initial input c_t as auxiliary conditioning signals.

The two-stage training algorithm is summarized in Algorithm 1.

Sampling Process

The generation of cascades proceeds through a time-aligned reverse denoising process, governed by the recursive formulation:

$$p_\theta = \begin{cases} F_\theta(\mathbf{s}^{(n)}, i_n) & \text{if } n = N-1, \\ \mathcal{I}_\phi(\mathbf{c}_t, F_\theta(\mathbf{s}^{(n)}, i_n), i_{n+1}) & \text{otherwise,} \end{cases} \tag{14}$$

with the initialization $\mathbf{s}^{(0)} := \mathbf{c}_t$, and intermediate states $\mathbf{s}^{(n)} \approx \mathbf{c}_{t+i_n}$ representing progressively refined predictions. To synchronize the generative procedure with the temporal evolution of the underlying cascade, the diffusion steps are indexed in reverse chronological order. Specifically, $n = 0$ corresponds to the initial observation c_t , while $n = N$ denotes the final target prediction c_{t+h} .

At a high level,let i be the time variable, ours diffusion model models the cascade dynamics as

$$\frac{d\mathbf{c}(i)}{di} = \frac{d\mathcal{I}_\phi\left(\mathbf{c}_t, F_\theta(\mathbf{c}, i), i\right)}{di}. \tag{15}$$

then this forward progression enable the generation module to operate with fewer diffusion steps and reduced data requirements.

3.3 Prediction Module

Based on the cascade representation obtained after h timesteps, denoted as c^i_{t+h}, we pass it through a Multi-Layer Perceptron (MLP) to predict the incremental popularity:

$$\widehat{\Delta P} c^i = f(\mathbf{c}t + h^i), \tag{16}$$

where $f(\cdot)$ represents the MLP functions.

To optimize the model, we employ the Mean Squared Logarithmic Error (MSLE) as the loss function, which is defined as:

$$\mathcal{J}(\theta) = \frac{1}{n} \sum_c \left(\log(\Delta P_{c^i}) - \log(\widehat{\Delta P_{c^i}}) \right)^2, \tag{17}$$

where n denotes the number of training cascades.

4 Experiment

In this section, we perform experiments on three datasets to assess the efficacy of our approach.

4.1 Experimental Setup

This section describes the dataset, evaluation metrics, baselines, and implementation details of our experiments.

Datasets.

We use three datasets, frequently employed in information propagation studies, derive from social media platforms and academic citation networks:

- **Twitter** [Weng *et al.*, 2013]dataset captures information cascades spanning from March 24 to April 25, 2012, where each cascade illustrates the dissemination of a hashtag via retweets.
- **Weibo** [Cao *et al.*, 2017]dataset includes information cascades collected on 1 July 2016, each cascade representing the spread of a post through retweets.
- **APS** dataset comprises information cascades up to the year 2017, each cascade reflecting the citation trajectory of a research paper.

Following the approach of CTCP[Lu *et al.*, 2023], we randomly select 70%, 15%, and 15% of the cascades for training, validation, and testing, respectively. For data preprocessing, we set the observation window of a cascade to 1 day on Twitter, 30 minutes on Weibo, and 2 years on APS. Notably, for Weibo and Twitter, this observation period is half of what was used in prior studies. We then predict the increment in cascade popularity from the observation time to the last recorded cascade instance. This setup for prediction timepoints also adheres to methodologies established in previous work.

Evaluation Metrics

We employ four extensively recognized metrics to assess the performance of the comparative methods: Mean Squared Logarithmic Error (MSLE), Mean Absolute Logarithmic Error (MALE), Mean Absolute Percentage Error (MAPE), and Pearson Correlation Coefficient (PCC). These metrics serve distinct evaluative purposes: MSLE, MAPE, and MALE quantify the prediction error relative to the ground truth from various perspectives, while PCC gauges the correlation between the predicted values and the ground truth.

Baselines

We evaluate the performance of our approach against the following advanced baselines:

- **DeepHawkes** [Cao *et al.*, 2017] models each cascade as a set of diffusion pathways across users, employing a Gated Recurrent Unit (GRU) to capture the sequential progression of cascades.
- **MS-HGAT** [Sun *et al.*, 2022] constructs a sequence of temporally-sampled hypergraphs that encapsulate multiple cascades and users, leveraging hypergraph learning to compute cascade representations.
- **CasCN** [Chen *et al.*, 2019] frames each cascade as a temporal graph sequence, utilizing a combination of Graph Neural Networks (GNN) and Long Short-Term Memory (LSTM) networks to learn robust cascade representations.
- **TempCas** [Tang *et al.*, 2021b] integrates a specialized sequence modeling technique designed to capture overarching temporal patterns, complementing its learning on the cascade graph.
- **CasFlow** [Xu *et al.*, 2021] first derives user representations from both the social network and the cascade graph, and then employs a GRU in conjunction with a Variational AutoEncoder (VAE) to encode cascade representations.
- **CTCP** [Lu *et al.*, 2023], the state-of-the-art method for cascade prediction, groups multiple cascades by their shared propagation users, enabling a unified temporal and structural learning process across the cascades.

Model	Twitter				Weibo				APS			
	MSLE	MALE	MAPE	PCC	MSLE	MALE	MAPE	PCC	MSLE	MALE	MAPE	PCC
DeepHawkes	9.1023	2.3551	0.4358	0.6630	3.1234	1.3482	0.3205	0.7102	2.5017	1.2894	0.3120	0.5945
CasCN	8.2649	2.2509	0.4102	0.6904	2.9951	1.2730	0.3107	0.7254	2.4208	1.2419	0.3059	0.6090
MS-HGAT	8.2812	2.1904	0.3989	0.7025	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
TempCas	8.2043	2.1750	0.3994	0.6958	2.8542	1.2408	0.3019	0.7323	2.3641	1.2274	0.2980	0.6164
CasFlow	8.1045	2.1603	0.3950	0.7038	**2.7984**	1.2395	0.2987	**0.74351**	2.3315	1.2155	0.2963	0.6220
CTCP	8.5126	2.2802	0.4074	0.6962	2.9182	1.2356	**0.2972**	0.7319	2.3364	1.2173	0.2974	0.6126
DGPA(ours)	**8.0916**	**2.0668**	**0.3562**	**0.8136**	2.8329	**1.2014**	0.2975	0.7337	**2.3147**	**1.1906**	**0.2711**	**0.7048**

Table 1: Performance of all methods in three datasets, where the methods can be divided into two categories: feature-based, deep-learning-based methods from top to bottom in the table. The best results appear in bold and OOM indicates the out-of-memory error.

4.2 Overall Performance

Table 1 details the comparative performance of various models across three datasets: Twitter, Weibo, and APS. Several important conclusions can be drawn from these results.

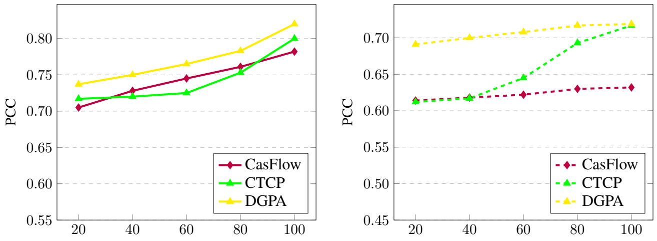
Firstly, it is evident that feature-based models such as DeepHawkes consistently lag behind other approaches across all evaluated metrics. This underperformance can be attributed to the inherent limitations of feature-based models in capturing the complex, nonlinear evolution patterns of cascade size, particularly during the early stages of information propagation.

We discover graph-based models generally outperform their sequence-based counterparts, emphasizing the critical role of incorporating both structural and temporal information within cascade graphs. For instance, CasFlow and CTCP demonstrates good performance, particularly on the Weibo dataset, suggesting its effectiveness in modeling both the temporal and structural dynamics of information spread.But their performance is not excellent over shorter observation periods.

The superior performance of DGPA across metrics, especially on datasets like Twitter and Weibo where the observation windows are inherently shorter, underscores its capability to generate accurate predictions with minimal initial data.

4.3 Sensitivity to Observation Time

In the experiments conducted on the Twitter and APS datasets, we selected observation periods corresponding to the 0th to 20th, 20th to 40th, 40th to 60th, 60th to 80th, and 80th to 100th percentiles. We then plotted the performance of the top three models during these intervals. As shown in Figure 3. Across both datasets (Twitter and APS), all models show an improvement in PCC as the observation window extends. This is expected because which allows models to better capture the underlying dynamics of information propagation. The CTCP model exhibits a particularly pronounced improvement in PCC as the observation window increases. This suggests that CTCP benefits significantly from additional data, likely because its prediction mechanism heavily relies on the volume of observed data to identify and exploit correlations between different cascades. However, at shorter observation windows, CTCP's performance lags behind. The CasFlow model shows a more modest improvement as the observation window increases, particularly on the APS dataset where its performance plateaus earlier compared to the other models. This could imply that CasFlow's ability to capture cascade dynamics is somewhat limited, especially in scenarios where early prediction is crucial. The model's reliance on early-stage data might not be as strong as DGPA's. The DGPA model maintains higher PCC values across all observation periods. This indicates that DGPA is particularly robust to varying lengths of observation windows. Notably, DGPA significantly outperforms the other models during the early stages of information cascades (i.e., shorter observation windows). This early-stage advantage suggests that DGPA is adept at capturing the initial dynamics of information spread, possibly due to its generative nature, which allows it to model cascade dynamics more effectively even with sparse data.



(a) Twitter

(b) APS

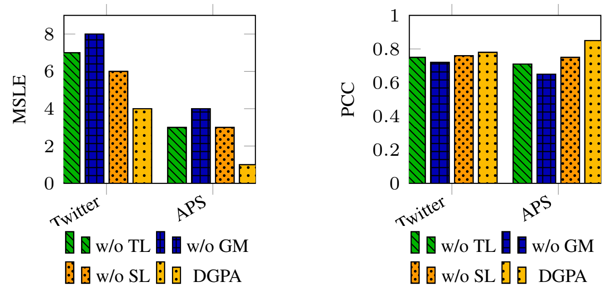
Figure 3: Observation time sensitivity analysis.

4.4 Ablation Study

We compare DGPA with the following variations on Twitter and APS to investigate the contribution of submodules to the prediction performance.

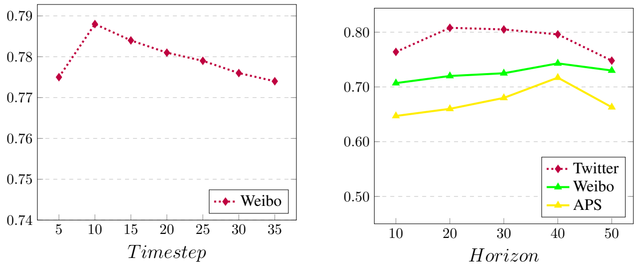
- **w/o TL** removes the temporal learning module.
- **w/o SL** removes the structural representation of users.
- **w/o GM** removes the cascade generation module.

From Figure 4, we can observe the following: Firstly, The removal of the temporal learning (TL) module leads to a significant drop in both MSLE and PCC, particularly on the Twitter dataset. This suggests that temporal dynamics play a crucial role in the early stages of information dissemination, where the timing and pace of retweets or citations are key indicators of future cascade popularity. The temporal



(a) MSLE on Twitter/APS (b) PCC on Twitter/APS

Figure 4: Ablation study on Twitter and APS.



(a) PCC vs. $Timestep$ (b) PCC vs. $Horizon$

Figure 5: Hyperparameter sensitivity analysis.

features captured by this module are especially vital in scenarios where observation periods are short, such as in social media environments. Secondly, the structural representation module (SL), while still contributing to the model's overall performance, has a less pronounced impact compared to the TL module, especially on the APS dataset. In networks like APS, where structural changes occur more slowly, temporal evolution may play a more significant role in predicting future cascade growth. The cascade generation module stands out as the most critical component of the DGPA model. The cascade generation module's ability to simulate future cascade propagation based on limited initial data is vital for long-term predictions. Its significant contribution to both MSLE and PCC metrics highlights the importance of generating synthetic data.

4.5 Hyperparameter Sensitivity Analysis

Here, we conduct a hyper-parameter sensitivity analysis on two parameters: the length of the $Timestep$, representing the interval of the cascade, and the number of iterations during the sampling process, denoted as $Horizon$. In Figure 5a, considering the rapid short-term propagation and the evident cascade changes on Weibo, we observe that as $Timestep$ increases, the model performance initially improves but then gradually declines.A larger number of diffusion steps require more frequent sampling, yet if $Timestep$ is too large, the generated embeddings may not align with the real cascade evolution distribution. Figure 5b illustrates that as the number of iterations required for sampling increases, model performance gradually improves and then declines. This indicates that an excessive number of iterations may introduce noise, thereby diminishing the model's effectiveness.

5 Related Work

Cascade Popularity Prediction. Cascade popularity prediction aims to forecast the future size of information cascades. Early approaches relied on manually engineered features, such as content attributes and user profiles [Cheng *et al.*, 2014],[Szabo and Huberman, 2010; Li *et al.*, 2017], but these methods required significant human input and lacked generalizability. Sequence-based models later conceptualized cascades as diffusion sequences, capturing their temporal dynamics. For example, Cao et al. [Cao *et al.*, 2017; Zhao *et al.*, 2021] employed Gated Recurrent Units (GRUs) to derive path-level representations, which were aggregated into a unified cascade representation. However, these methods often overlooked the structural intricacies inherent in cascades. Recent advancements introduced graph-based approaches that frame cascades as dynamic graphs, leveraging graph representation learning to encode both temporal and structural features [C *et al.*, 2017], [X *et al.*, 2019], [X *et al.*, 2021]. Despite their effectiveness, these models typically treat cascades in isolation, ignoring interdependencies between them.

Diffusion Models. Denoising diffusion probabilistic models (DDPMs) have gained significant attention for their ability to model complex distributions and generate high-fidelity data [Ho *et al.*, 2020; Zhang *et al.*, 2023]. DDPMs operate via a forward process, where Gaussian noise is progressively added to the data, and a reverse process, where the model iteratively denoises the data to reconstruct it. This process enables DDPMs to capture intricate data structures and has proven effective in high-dimensional generative tasks.

In the context of information diffusion, DDPMs offer a novel approach for simulating the spread of information within social networks. By interpreting user interactions as sequences of noisy observations, the reverse diffusion process generates plausible propagation pathways, capturing both temporal sequences and structural characteristics of network interactions. This methodology provides deeper insights into the mechanics of information dissemination and social network dynamics.

6 Conclusion

In this work, we introduced DGPA (Diffusion Guided Propagation Augmentation), a novel generative framework for early-stage information popularity prediction. By integrating a time-conditioned interpolation mechanism in the forward diffusion process, DGPA generates continuous and temporally coherent cascade representations from sparse data. In the reverse process, DGPA aligns generated representations with specific timestamps, addressing challenges such as irregular sampling and the difficulty of simulating real-world propagation dynamics. Experiments on real-world datasets demonstrate that DGPA significantly outperforms state-of-the-art methods in early-stage popularity prediction tasks.

Acknowledgements

This work is supported by the National Key Research and Development Program of China (No.2023YFB3107000)

References

[C *et al.*, 2017] Li C, Ma J, and et al. Guo X. Deepcas: An end-to-end predictor of information cascades. In *Proceedings of the 26th International Conference on World Wide Web*, pages 577–586. ACM, 2017.

[Cao *et al.*, 2017] Qi Cao, Huawei Shen, Keting Cen, Wentao Ouyang, and Xueqi Cheng. 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)*, pages 1149–1158. ACM, 2017.

[Cao *et al.*, 2019] Jiezhang Cao, Langyuan Mo, Yifan Zhang, Kui Jia, Chunhua Shen, and Mingkui Tan. Multi-marginal wasserstein gan. *Advances in Neural Information Processing Systems*, 32, 2019.

[Chen *et al.*, 2019] Xueqin Chen, Kunpeng Zhang, Fan Zhou, Goce Trajcevski, Ting Zhong, and Fengli Zhang. Information cascades modeling via deep multi-task learning. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 885–888. ACM, 2019.

[Chen *et al.*, 2022] Xi Chen, Xiangmin Zhou, Jeffrey Chan, Lei Chen, Timos Sellis, and Yanchun Zhang. Event popularity prediction using influential hashtags from social media. *IEEE Transactions on Knowledge and Data Engineering*, 34(10):4797–4811, 2022.

[Cheng *et al.*, 2014] Justin Cheng, Lada A Adamic, P Alex Dow, Jon M Kleinberg, and Jure Leskovec. Can cascades be predicted? In *Proceedings of the 23rd international conference on World wide web*, pages 925–936. ACM, 2014.

[Cheng *et al.*, 2024] Zhangtao Cheng, Fan Zhou, Xovee Xu, Kunpeng Zhang, Goce Trajcevski, Ting Zhong, and S Yu Philip. Information cascade popularity prediction via probabilistic diffusion. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

[Ho *et al.*, 2020] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

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

[Lee *et al.*, 2021] Kwonjoon Lee, Huiwen Chang, Lu Jiang, Han Zhang, Zhuowen Tu, and Ce Liu. Vitgan: Training gans with vision transformers. *arXiv preprint arXiv:2107.04589*, 2021.

[Leskovec *et al.*, 2007] Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1):5–es, 2007.

[Li *et al.*, 2017] Chaozhuo Li, Senzhang Wang, Dejian Yang, Zhoujun Li, Yang Yang, Xiaoming Zhang, and Jianshe Zhou. Ppne: property preserving network embedding. In *Database Systems for Advanced Applications: 22nd International Conference, DASFAA 2017, Suzhou, China, March 27-30, 2017, Proceedings, Part I 22*, pages 163–179. Springer, 2017.

[Li *et al.*, 2018] Chaozhuo Li, Senzhang Wang, Philip S Yu, Lei Zheng, Xiaoming Zhang, Zhoujun Li, and Yanbo Liang. Distribution distance minimization for unsupervised user identity linkage. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 447–456, 2018.

[Li *et al.*, 2021] Chaozhuo Li, Bochen Pang, Yuming Liu, Hao Sun, Zheng Liu, Xing Xie, Tianqi Yang, Yanling Cui, Liangjie Zhang, and Qi Zhang. Adsgnn: Behavior-graph augmented relevance modeling in sponsored search. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 223–232, 2021.

[Lu *et al.*, 2023] Xiaodong Lu, Shuo Ji, Le Yu, Leilei Sun, Bowen Du, and Tongyu Zhu. Continuous-time graph learning for cascade popularity prediction. In Edith Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 2224–2232. International Joint Conferences on Artificial Intelligence Organization, 8 2023.

[Sun *et al.*, 2022] Ling Sun, Yuan Rao, Xiangbo Zhang, Yuqian Lan, and Shuanghe Yu. Ms-hgat: Memory-enhanced sequential hypergraph attention network for information diffusion prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4156–4164. AAAI, 2022.

[Szabo and Huberman, 2010] Gabor Szabo and Bernardo A Huberman. Predicting the popularity of online content. *Communications of the ACM*, 53(8):80–88, 2010.

[Tang *et al.*, 2021a] Chen Tang, Wei Zhan, and Masayoshi Tomizuka. Exploring social posterior collapse in variational autoencoder for interaction modeling. *Advances in Neural Information Processing Systems*, 34:8481–8494, 2021.

[Tang *et al.*, 2021b] Xiangyun Tang, Dongliang Liao, Weijie Huang, Jin Xu, Liehuang Zhu, and Meng Shen. Fully exploiting cascade graphs for real-time forwarding prediction. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*, pages 582–590. AAAI Press, 2021.

[Weng *et al.*, 2013] Lilian Weng, Filippo Menczer, and Yong-Yeol Ahn. Virality prediction and community structure in social networks. *Scientific Reports*, 3(1):1–6, 2013.

[X *et al.*, 2019] Chen X, Zhou F, and et al. Zhang K. Information diffusion prediction via recurrent cascades convolution. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 770–781. IEEE, 2019.

[X *et al.*, 2021] Xu X, Zhou F, Zhang K, and et al. Casflow: Exploring hierarchical structures and propagation uncertainty for cascade prediction. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3484–3499, 2021.

[Xu *et al.*, 2021] Xovee Xu, Fan Zhou, Kunpeng Zhang, Siyuan Liu, and Goce Trajcevski. Casflow: Exploring hierarchical structures and propagation uncertainty for cascade prediction. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[Zhang *et al.*, 2023] Peiyan Zhang, Jiayan Guo, Chaozhuo Li, Yueqi Xie, Jae Boum Kim, Yan Zhang, Xing Xie, Haohan Wang, and Sunghun Kim. Efficiently leveraging multi-level user intent for session-based recommendation via atten-mixer network. In *Proceedings of the sixteenth ACM international conference on web search and data mining*, pages 168–176, 2023.

[Zhao *et al.*, 2019] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Balancing learning and inference in variational autoencoders. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 5885–5892, 2019.

[Zhao *et al.*, 2021] Jianan Zhao, Chaozhuo Li, Qianlong Wen, Yiqi Wang, Yuming Liu, Hao Sun, Xing Xie, and Yanfang Ye. Gophormer: Ego-graph transformer for node classification. *arXiv preprint arXiv:2110.13094*, 2021.