mathematics

MDPI

Article

Predicting Popularity of Viral Content in Social Media through a Temporal-Spatial Cascade Convolutional Learning Framework

Zhixuan Xu [1] and Minghui Qian [2,*]

[1] College of Business and Administration, Capital University of Economics and Business, Beijing 100071, China; zhixuan@cueb.edu.cn
[2] School of Information Resource Management, Renmin University of China, Beijing 100872, China
* Correspondence: qmh@ruc.edu.cn

Abstract: The viral spread of online content can lead to unexpected consequences such as extreme opinions about a brand or consumers' enthusiasm for a product. This makes the prediction of viral content's future popularity an important problem, especially for digital marketers, as well as for managers of social platforms. It is not surprising that conventional methods, which heavily rely on either hand-crafted features or unrealistic assumptions, are insufficient in dealing with this challenging problem. Even state-of-art graph-based approaches are either inefficient to work with large-scale cascades or unable to explain what spread mechanisms are learned by the model. This paper presents a temporal-spatial cascade convolutional learning framework called ViralGCN, not only to address the challenges of existing approaches but also to try to provide some insights into actual mechanisms of viral spread from the perspective of artificial intelligence. We conduct experiments on the real-world dataset (i.e., to predict the retweet popularity of micro-blogs on Weibo). Compared to the existing approaches, ViralGCN possesses the following advantages: the flexible size of the input cascade graph, a coherent method for processing both structural and temporal information, and an intuitive and interpretable deep learning architecture. Moreover, the exploration of the learned features also provides valuable clues for managers to understand the elusive mechanisms of viral spread as well as to devise appropriate strategies at early stages. By using the visualization method, our approach finds that both broadcast and structural virality contribute to online content going viral; the cascade with a gradual descent or ascent-then-descent evolving pattern at the early stage is more likely to gain significant eventual popularity, and even the timing of users participating in the cascade has an effect on future popularity growth.

Keywords: viral spread; information cascade; graph learning; popularity prediction

MSC: 68M11

check for updates

Citation: Xu, Z.; Qian, M. Predicting Popularity of Viral Content in Social Media through a Temporal-Spatial Cascade Convolutional Learning Framework. *Mathematics* **2023**, *11*, 3059. https://doi.org/10.3390/math11143059

Academic Editors: Shuo Yu and Feng Xia

Received: 8 June 2023
Revised: 2 July 2023
Accepted: 7 July 2023
Published: 11 July 2023

1. Introduction

The viral spread of online content, which is also known as electronic word-of-mouth, viral marketing, or information cascade, is generally understood as the rapid growth of popularity/cascade size through individual-to-individual information sharing processes [1,2]. As information efficiency is greatly enhanced by social media, online content on Twitter, Facebook, or Weibo can go viral very quickly [3], and sometimes even cause an extremely favorable or disastrous consequence in a very short period. For example, in 2012, a 466-word post by a disgruntled customer in Odeon Cinemas' Facebook brand community maliciously slandered the brand. This post went viral in only a few h and prompted more than 94,000 likes at the end, which severely damaged the brand's reputation and made it lose thousands of customers [4]. Since the viral spread of online content, especially those with massive cascade sizes has a significant impact on other consumers' brand attitude [5] as well as product sales [6], predicting the future popularity (or more precisely, the final cascade size in a social network) is greatly valuable for managers to make strategic decisions or take precautions before the unexpected results [4,7,8]. Unfortunately, on the one hand, the actual mechanisms of online viral spread are still inconclusive [2,9]. On the other hand, conventional prediction methods, such as feature-based approaches and generative approaches, are inadequate to cope with the rapid and complex information spread in social media.

Historically, researchers firstly rely on hand-crafted features to build explainable models for popularity prediction [10–12]. Although most feature-based models have yielded considerably competitive results, they are still not widely used in real applications since some of the features are either unavailable (user features, browsing histories, etc.) due to privacy concerns or cannot be generalized to different scenarios (micro-blogs, music, videos, etc.) [9]. Some researchers also proposed generative approaches that regard the information retweeting/sharing of consumers as event sequences in the continuous temporal domain and predict the future popularity by modeling its temporal dynamics [13–15]. However, almost all generative models are built on strong assumptions of viral spread mechanisms, such as the uniform contribution of each new retweet for future popularity growth [14], leaving huge gaps between the predicted results and the actual growth sizes. Recently, graph learning algorithms that have been successfully applied in chemical structure classifications and traffic predictions have shed new light on the popularity prediction of online content. Several graph-based approaches are developed to make predictions through informative structural features that are unsupervisedly extracted from the information diffusion network/cascade graph. While the proposed graph-based approaches (e.g., DeepCas [16], DeepHawkees [17], CasCN [18], etc.), which do not rely on user characteristics or content features, are widely applicable to a variety of information spread scenarios, the extant research still is yet to address the following challenges. First, due to the power-law distribution of the online content popularity, some cascades can reach a considerable size at the early stage; thus, handling those oversize cascades is one of the main obstacles encountered in real-life applications. Second, the end-to-end prediction manners and the graph convolutions defined in the Fourier domain are helpless for understanding the actual viral spread mechanism in social media. Lastly, since common graph learning algorithms are originally designed for embedding structural features of nodes or graphs, it is a challenge for a graph-based model to extract the spreading dynamics directly from a cascade graph as well as to appropriately incorporate the temporal information into the model.

Our research proposes a spatial-temporal cascade convolution learning framework called ViralGCN in response to these challenges. As shown in Figure 1, our framework comprises 4 parts; it first starts with an adaptive node-sampling process to select a certain number of nodes/users from the oversize cascade graph, followed by several bi-directional spatial convolutional layers that extract the local structural features of each node from both directions of information inflow and outflow; then, a temporal information aggregation layer is incorporated to capture the spread dynamics as well as the time decay effects, and finally an MLP layer makes the prediction. Compared to extant popularity prediction methods, ViralGCN offers the following new capabilities:

1. Instead of calculating the whole cascade graph, we propose an adaptive node-sampling process to input sufficient information of large-scale cascades, where to avoid the omission of important structural information, the node with higher degrees (i.e., carrying more information) has a higher weight to be sampled.
2. The bi-directional spatial convolutional layer allows us to obtain each node's representation, which contains structural information from both directions of information inflow and outflow, making it more helpful to explain which features of users are extracted as well as to understand the actual mechanisms of online viral spread.
3. Our proposed temporal information aggregation layer provides an innovative way to capture the spread dynamics directly from a cascade graph where the obtained nodes' representations are aggregated according to each divided time window, and it

provides another example of the combination of structural and temporal information in a graph-based popularity prediction model.

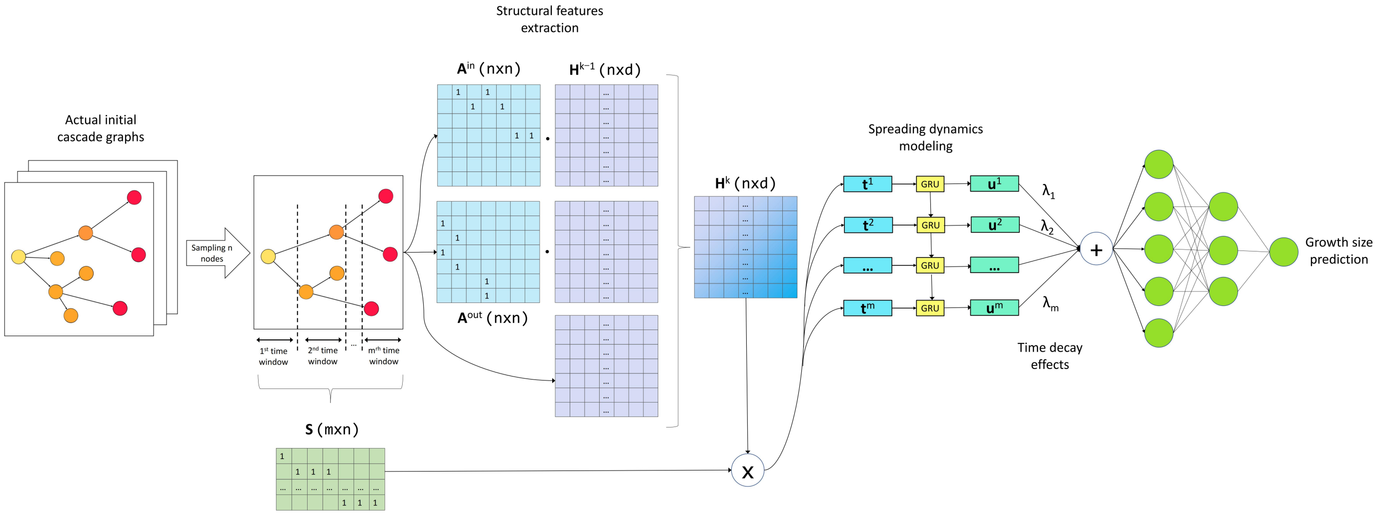


Figure 1. The framework of ViralGCN.

We conduct experiments on a real-world viral spread dataset, which is crawled from famous social media platform Weibo and widely used in recent popularity prediction studies [17,18]. The empirical investigation demonstrates that our proposed ViralGCN remarkably outperforms not only the conventional feature-based models but also the state-of-art graph-based models. In addition, to verify the effectiveness of ViralGCN's components, we also conduct an ablation experiment by developing several variants of the proposed model. The results show that ViralGCN achieves a significant improvement in computation efficiency, but its prediction accuracy is still comparable with the variant model that does not use the node sampling method. Other modified models show performance degradation at different levels compared with the original ViralGCN.

Moreover, this paper also tries to provide some insights into the actual viral spread mechanisms from an artificial intelligence perspective and help managers' decision-making. The method for visualizing high-dimensional data is utilized to explain which structural and temporal features are extracted and how these features affect future popularity growth. Consequently, we find 3 main evolving patterns of early viral spread in which the gradual descent and the ascent-then-descent evolving pattern are more likely to gain large final popularity. In addition, the visualization of nodes' embeddings helps us identify 4 types of users (i.e., broadcasters, influential disseminators, active responders, and responder's responders), which is helpful for managers to tailor their strategies to specific types of users. Furthermore, in response to the argument of whether broadcast diffusion or word-of-mouth drives the popularity of online content, we find that both broadcast and structural virality contribute to online content going viral. Moreover, the learned time decay effects indicate that the timing of consumers' retweeting/sharing also affects future popularity growth. Overall, our empirical studies not only suggest that the proposed approach complements existing graph-based prediction methods but also provide a promising new way to explain the mechanisms of viral spread in social media.

In the rest of the paper, Section 2 makes a literature review of related works. In Section 3, we define the problems and provide a specific illustration of our method. The results of performance comparisons and ablation experiments are presented in Section 4. We also make a study on what features are extracted and on the nature of the relationship between the learned features and popularity growth in Section 4. The last Section 5 concludes the paper and introduces future work.

2. Related Works

Perspectives of prediction. As the viral spread of online content often causes significant consequences in social media, predicting whether an advertisement or eWOM can go viral attracts significant attention from both managers and researchers [4]. Some researchers in the management domain try to figure out what drives the popularity of online content from a micro-level perspective, where the subjective factors such as consumers' emotions aroused by the content [19,20] and the social ties between consumers [21] are thought to influence individuals' sharing behaviors. Although the micro-level methods are effective at predicting the activation of a specific individual, they are inefficient at predicting the collective behaviors of a group of people (e.g., the future popularity of viral content) due to the individual heterogeneity as well as the inaccessibility of consumer's actual feelings [8]. On the contrary, the macro-level methods, which regard viral spread as the growth of a cascade in the social network, are generally adopted by current researchers.

Features for prediction. One of the key issues for building a macro-level model is to make sure which features can provide predictive information. The structural features of the initial cascade graph (node degrees, edge density, diffusion depth, etc.) have been proven to play an important role in popularity prediction by many researchers [12,22,23]. Specifically, the authors of [24] conducted a study on microblog diffusion networks and found that regarding the initial cascade with a lower edge density but a higher diffusion depth, diverse early adopters are more likely to go viral. On the other hand, Ref. [2] argues against this opinion by analyzing billions of online diffusion events and finding that a lot of cascades with relatively low structural virality can still obtain considerable attention through broadcasting, in which only several influential opinion leaders produce popularity growth. Although the researchers disagree on what type of initial cascade structure could bring higher popularity, they both provide strong evidence of predictive information underlying the structural features of the initial cascade graph. For instance, the degree of an early adopter node implies its potential influence, and the diffusion depth reveals the cascade virality.

Temporal features of spreading (mean arriving time [12], mean reaction time [22], evolving patterns [22], etc.) are also considered valuable in popularity prediction. For example, Ref. [25] finds a strong relationship between the cascade's early popularity and its final popularity. Ref. [26] indicates the existence of a log-norm distribution in users' reactions to a new post tweet. Additionally, some researchers modeling users' retweet/sharing time series also find the rich-get-richer and the time decay effect in cascades' evolving patterns [27,28]. Even in the medical field, temporal features are used to build deep learning models for virus transmission prediction [29]. Since there is much evidence for the effect of a cascade's dynamics as well as its initial structural features on future popularity, our model must make predictions by combining both features.

Except for the structural and the temporal features, the content features (topics [30], sentiments [31], hashtags [32], etc.) are also widely adopted by extant studies as well. However, the efficacy of content features in predicting popularity is still controversial. Ref. [12]'s empirical study shows that the content features are less important when the cascade's size grows larger. The conclusion is consensus with the herding theory [33] and the informational cascade theory [34], where individuals' decision-making is significantly influenced by others' behaviors instead of their own knowledge. In addition, many researchers also find that there can be a huge distance in terms of final popularity between two examples of identical online content [35–37]. Moreover, another limitation of content features is that they are not generalizable for different viral spread scenarios. For example, the semantic features discussed above cannot be applied to predict the popularity of images [38] or videos [39]. Therefore, considering the effectiveness of the content features as well as the model generalizability, we do not incorporate the content features in the proposed model.

Methods for prediction. The extant methods for the prediction can be roughly divided into three categories (e.g., the feature-based, the generative, and the graph-based approaches). The feature-based approaches are the most common in conventional research, which usually builds a regression or classification model by a bunch of well-designed features [11,12,40,41]. The limitations of these feature-based approaches are apparent. First of all, some extracted features such as the content features are too specific to the particular type of information [32]. In addition, the current hand-craft features (degrees, centrality, border edges, etc.) are insufficient to reveal the whole topological characteristics of a graph. For instance, two nodes in a graph with the same degree of centrality may have different structures, limiting the exploration of unknown spread mechanisms of online content. Another type of cascade-prediction method, the generative approach, focuses more on cascades' evolving patterns and makes predictions by modeling the dynamics of spreading. For example, Ref. [28] builds a time-relaxation function and a reinforcement function to depict the decay effect and the rich-get-richer effect of cascade growth. Ref. [27] employs the self-exciting point process model to simulate the spreading dynamics of tweets. However, due to the cascade evolution mechanism remaining elusive, the models have to be built based on strong assumptions and oversimplifications of reality. As a result, there is often a big gap between the model predictions and the actual popularity in real applications [9].

Recently, regarding the unsupervised data input and excellent nonlinear modeling capabilities, the graph-based approaches combining both graph and deep learning methods attract more attention in the popularity prediction area. Ref. [16] proposes the first end-to-end cascade prediction system (DeepCas) that learns the representation of each cascade graph through a series of node sequences sampled by random walks. Another model called DeepHawkes [17] combines deep learning with the Hawkes process and considers the time decay effect of each diffusion path. Ref. [18] proposes CasCN, which defines the spectral convolution of cascade graphs and adopts a recurrent neural network to capture the dynamics of spreading. DMT-LIC [42] adopts a multi-layer graph attention network to embed the nodes in graphs and simply inputs each node to an LSTM by the order of retweeting where only partial spreading dynamics can be captured. We compare extant graph-based approaches in Table 1, in which the deficiencies of current models are apparent. First, there is a lack of solutions to handle the input of oversize cascade graphs. Additionally, some models partially or completely fail to take into account the temporal information of viral spread, which has been proven valuable for popularity prediction. Moreover, the outputs of the graph-learning method are mainly elusive graph embeddings that are helpless to understand the actual viral spread mechanisms. Thus, in this study, we try to propose an innovative graph learning framework ViralGCN to address these challenges.

Graph neural networks. Graph neural networks (GNNs) are motivated by the standard convolutional neural networks (CNNs) that use a shared filter to extract the localized spatial features and compose them to construct highly expressive representations [43]. Similar to the images, the whole representation of a graph can also be obtained by assembling all localized structural features. However, the standard CNNs cannot directly operate on non-Euclidean data, which usually relies on preprocessing work (e.g., random walk) to transform the graph to regular Euclidean data (node sequences). As a result, the conventional graph representation approaches, such as Deepwalk [44], Node2Vec [45], and Struc2Vec [46], cannot deal with dynamic graphs, and non-shared parameters lower computation efficiency as well.

Therefore, the GNNs are developed as a generalization of CNNs to graphs and inherit the advantages of CNNs. For example, the GNN proposed in [47] is the first deep learning method that directly processes graph data and makes embeddings for nodes by aggregating the information of their neighbors. Following the GNN, many variants of GNN are developed in the following studies. For example, the DGP [48] extends the GNN for directed graphs; the G2S [49] incorporates the edge information into the model; and the GGNN [50] combines GRU with the update process of the node hidden state.

Table 1. The comparison of extant graph-based prediction models.

Models	The Method to Extract the Structural Information of a Cascade Graph	The Method to Deal with the Oversize Cascade Graphs	The Method to Extract the Temporal Information of Spreading	The Temporal Information Considered by the Model	The Explanation of Learned Popularity Growth Mechanisms
DeepCas [16]	Transform the graph into a set of node sequences by random walk	Sample a certain number of sequences that carry sufficient information of a cascade graph			
DeepHawkes [17]	Transform the graph into a set of diffusion paths		Give a time decay effect for each observation time window	Time decay effect	
CasCN [18]	Obtain the representations of cascade graphs by computing their Laplacian matrices		Input the sub-graph of each observation time window to an LSTM, and the time decay effects are given	Time decay effect and spread dynamics	
DMT-LIC [42]	Adopt a multi-layer graph attention network to embed the nodes in graphs		Embed the diffusion process by inputting nodes to an LSTM in time order	Partial spread dynamics	
The proposed VirGCN	Develop a bi-directional spatial graph convolution of cascade graph to extract nodes' local structural features	Sample a certain number of nodes that carry sufficient information of oversize cascade graphs	Aggregate the nodes of each observation time window, then adopt a GRU to capture the spread dynamics; time decay effects are given	Time decay effect and spread dynamics	Explain the extracted features of each node as well as the time decay effects; explore the effect of early evolving patterns and cascade structures on future popularity growth

In general, the current GNNs can be divided into two categories according to convolution approaches: the spectra methods and spatial methods. In a spectra method, the convolution operation is defined in the Fourier domain by computing the eigendecomposition of the graph Laplacian [51]. However, the Fourier domain's convolution is unintuitive for people seeking to understand which features the model captures from data. In contrast, the spatial method operates convolution directly following information diffusion paths [52], which is also in accord with the real cascade evolution. Moreover, the spectral convolution is inflexible for large cascade learning as it usually requires the computing of the Laplacian of the whole graph, while the spatial convolution can be combined with the node sampling method to efficiently handle those oversize cascade graphs.

Overall, existing studies have adequately demonstrated the advantages of graph neural networks for online content popularity prediction, but conventional graph-based methods still need to address the following issues: first, how to efficiently handle cascade graphs with large scales, and additionally how to reveal the viral spread mechanisms learned by the models. Therefore, our work will deal with the above challenges by designing a spatial-temporal learning framework, which attempts to extract the information in large cascade graphs through an efficient node sampling algorithm and explore the diffusion mechanisms of viral content by visualizing the learned features of each node.

3. Method

In this study, we treat the viral spread of online content as the growth of information cascade graphs, where the structural and temporal information of the initial graphs are considered as the key factors that decide their growth sizes, i.e., their future popularity [12,22], since social media (Twitter, Weibo, etc.) usually precisely records who retweets a message, and when this happens, it is feasible for managers to observe a dynamic cascade graph at early stages and predict its future popularity. In this section, we first define the problem and then introduce how our proposed ViralGCN makes predictions. The notations used in this paper are shown in Table 2.

Table 2. Notations used in this paper.

Symbol	Description
G	A snapshot of the global social network.
C	A set of retweet cascades of viral content in G.
c	The retweet cascade of viral content.
g_c^t	The graph of cascade c within the time duration t after its origination.
V_c^t,E_c^t,T_c^t	The set of nodes, edges, and retweet timestamps in g_c^t.
Δt	The fixed time interval.
Δs_c	The popularity increment size of cascade c after Δt.
$\mathbf{R}_c$	The obtained representation of g_c^t.
$D(V)$	The degree of node v.
V_c^s	The set of sampled nodes in g_c^t.
$\mathbf{X}_c$	The stacked initial embedding of each node in V_c^s.
$\mathbf{A}_c^{in}$, $\mathbf{A}_c^{out}$	The adjacency matrices of g_c^t in distinct directions.
k	The fixed aggregation depth.
$\mathbf{H}_c^k$	The hidden states of g_c^t after k layers.
$\mathbf{h}_v^k$	The hidden states of node v after k layers.
m	The fixed number of divided time windows.
$\mathbf{S}_c$	The stacked retweet node sequences in each time window of g_c^t.
$\mathbf{u}_c^i$	The hidden state of ith time window.
λ_i	The decay effect of ith time window.
n	The number of sampled nodes.

3.1. Problem Definition

The core problem to be addressed in this study is how to predict the final popularity of viral online content at an early stage of spreading, and the following definitions have to be clarified before describing the problem by mathematical formulas.

3.1.1. Global Network

Suppose that at time t_0 we take a snapshot of a social network $G = (V, E)$, where V is the set of nodes in this network at t_0 and $E \subset V \times V$ is the set of edges between nodes. A node is a user in the social network, and an edge shows the relationship between two users.

3.1.2. Dynamic Cascade Graph

We denote a set of retweet cascades of viral content in the global social network G as C. Each cascade $c \in C$ with a duration t after its origination is described by a directed graph $g_c^t = (V_c^t, E_c^t, T_c^t)$, where V_c^t is a set of nodes that have been involved in the cascade c within duration t after the original post, a directed edge $(v_i, v_j) \in E_c^t$ represents that node v_j retweets the message from node v_i, and a timestamp $t_v \in T_c^t$ denotes the time elapsed between the original post and node v's retweet. Compared with previous definitions of the cascade graph [16,18], the timestamp t_v of each node that records when a user retweets the message is added to g_c^t. g_c^t could vary with the observation time t, so we call g_c^t, which includes temporal information of spreading the dynamic cascade graph.

3.1.3. Popularity Growth Size

In this study, the popularity/cascade size is defined as the number of retweets of a message. Due to the rich-get-richer phenomenon, there is usually an intrinsic correlation between the observed cascade size and its final size. In order to exclude its impact on our model, we predict the increment of a cascade's size after a given time interval Δt [16–18,42] instead of directly predicting its final size. Let $g_c^{t+\Delta t} = (V_c^{t+\Delta t}, E_c^{t+\Delta t}, T_c^{t+\Delta t})$ be the graph at time $t + \Delta t$. The popularity growth size can be denoted as $\Delta s_c = |V_c^{t+\Delta t}| - |V_c^t|$, and it is known that the $|V_c^{t+\Delta t}|$ is closer to the final size of a cascade when Δt is larger. Figure 2 gives an illustrative example to show the growth of a cascade graph, i.e., the spread of viral content in social media.

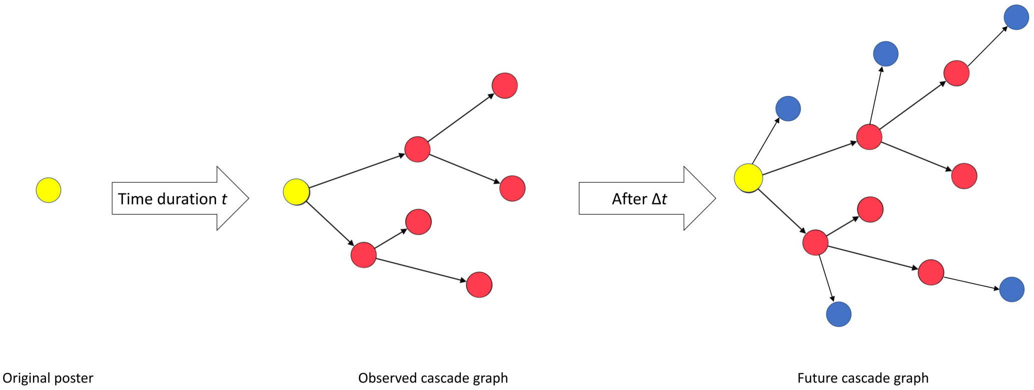


Figure 2. The spread of viral content in social media.

According to the framework of our model shown in Figure 1, we take the cascade graph at an early stage (i.e., the time duration t after posting) as the input, and the output is the predicted increment of the cascade size at time $t + \Delta t$. The model automatically embeds the structural information of nodes, temporal dynamics of spreading, and time decay effects for the final prediction. Let $\mathbf{R}_c$ be the obtained representation of the dynamic cascade graph g_c^t. Then, the future popularity prediction can be formulated as, given t, Δt, and $\{g_c^t\}_{c \in C}$, finding the optimal mapping function f that minimizes the following objective function:

$$O = \frac{1}{|C|} \sum_{c=1}^{|C|} (f(\mathbf{R}_c) - \Delta s_c)^2 \tag{1}$$

3.2. Model

The architecture of ViralGCN has been given in Figure 1. In the following, we explain its components. More details can be found at https://github.com/XUDZX/ViralGCN, accessed on 6 December 2021.

3.2.1. Node Embedding

We first generate an initial embedding for each node in the global graph G. Each node is represented as a one-hot vector $\mathbf{q} \in \mathbb{R}^{N_{node}}$, where N_{node} is the number of nodes. All of the nodes share an embedding matrix $\mathbf{A} \in \mathbb{R}^{d \times N_{node}}$ that transforms each node into its initial embedding vector $\mathbf{x} = \mathbf{Aq}$ such that $\mathbf{x} \in \mathbb{R}^d$.

3.2.2. Adaptive Nodes Sampling Process

The sizes of cascades in social media may vary in a huge range; however, the graph-learning models usually require a fixed input size. In contrast with the extant research that makes the input size of the model as large as possible, we develop an adaptive nodes sampling process to select a certain number of nodes that carry sufficient information of those oversize cascade graphs. As a sub-graph of the global social network, the cascade graph has scale-free property as well [53], where only a few nodes with high degrees play a key role in both information diffusion and graph structure maintenance. Therefore, to ensure the selected nodes carry sufficient information of a cascade graph, we apply the rule of sampling in Algorithm 1.

Algorithm 1 Node sampling algorithm

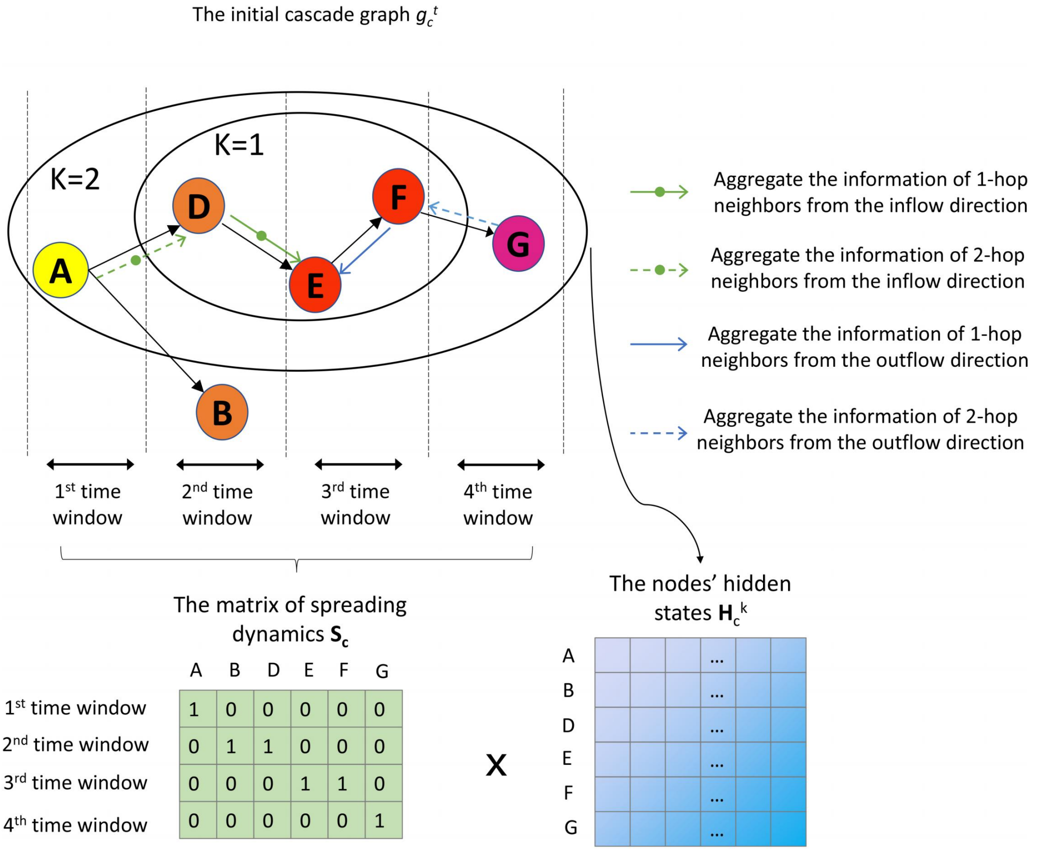
```
for $c \in C$ do
    $V_c^s \leftarrow \{\}$
    if $|V_c^t| < n$ then
        $V_c^s \leftarrow V_c^t$
        padding $V_c^s$ with 0 until $|V_c^s| = n$
    else
        while $|V_c^s| < n$ do
            random sample a node from $(V_c^t - V_c^s)$
            with the probability of $P(v) = \frac{D(v)}{\sum_{u \in (V_c^t - V_c^s)} D(u)}$
        end while
    end if
    return $V_c^s$
end for
```

It should be noted that $D(v)$ is the degree of node v. The probability is calculated to ensure that the selected nodes carry sufficient information about the cascade graph. Meanwhile, n is the number of nodes that need to be selected as well as the input size of the model. Since the best value of n could vary in different datasets, instead of setting a fixed value we make n a hyper-parameter that is trainable in our model. According to our algorithm, a few high-degree nodes that carry important information are more likely to be captured, and the value of n decides how many other nodes are enough to reveal the main structure of a cascade graph.

3.2.3. Bi-Directional Spatial Convolutional Layer

Since the structural features of the initial cascade graph can provide important clues about how the message will spread in the future, the main objective of this component is to extract the predictive structural information from the graph. Additionally, since the conventional graph representation approaches as well as the spectral graph convolution methods are less explainable, in this work, we are trying to develop a spatial cascade convolutional method not only to extract the structural information of cascades but also to explain which different roles users play in the viral spread by obtaining the representation of each node.

Bi-directional aggregation: Taking into account the information diffusion directions, we develop a bi-directional aggregation approach. As shown in Figure 3, when the model extracts the structural features of node E, green arrows with a circle and blue arrows represent that the model aggregates E's neighbors from the information inflow and outflow directions. As the viral spread of online content is a one-way process, distinguishing nodes from different directions provides more structural information of a cascade. For example, the edge directions let the model know that node A is the original poster, and node E retweets the message from node D. Therefore, we extract the cascade's structural information by generating two adjacent matrices $\mathbf{A}_c^{in} \in \{0,1\}^{n\times n}$ and $\mathbf{A}_c^{out} \in \{0,1\}^{n\times n}$ that, respectively, record each node's neighbors in distinct directions. Take a row of $\mathbf{A}_c^{out}$ as an example; the positions of its child nodes are 1, and the rest of the positions are 0, while in a row of $\mathbf{A}_c^{in}$, the positions of its parent nodes are 1, and the rest of the positions are 0.


The initial cascade graph g_c^t
K=2
K=1
A
B
D
E
F
G
Aggregate the information of 1-hop neighbors from the inflow direction
Aggregate the information of 2-hop neighbors from the inflow direction
Aggregate the information of 1-hop neighbors from the outflow direction
Aggregate the information of 2-hop neighbors from the outflow direction
1st time window
2nd time window
3rd time window
4th time window
The matrix of spreading dynamics $\mathbf{S_c}$
X
The nodes' hidden states $\mathbf{H_c}^k$
A
B
D
E
F
G
...

	A	B	D	E	F	G
1st time window	1	0	0	0	0	0
2nd time window	0	1	1	0	0	0
3rd time window	0	0	0	1	1	0
4th time window	0	0	0	0	0	1

Figure 3. The process of structural and temporal information extraction.

Aggregator function: As the localized structure of a node is determined by its neighbors, the main job of the cascade convolutional layer is to aggregate the information of each node's local neighbors. Due to the fact that the aggregator function should be symmetric (i.e., invariant for the input order of nodes) and trainable for maintaining high representational capacity, a mean aggregator is adopted in our model.

Aggregation depth: We also make k convolutional layers to extract the deeper structural information of each node. Figure 3 clearly illustrates how the process works. When $k = 1$, the node E embeds the information of its local neighbors into its hidden state $\mathbf{h}_E^1$ (the row of $\mathbf{H}_c^1$). Meanwhile, the updated hidden states of node F also include the information from its local neighbor, i.e., node G, which is one of the 2-hop neighbors of node E. After that, the information of node G will be captured by node E in the next layer. As a result, after k spatial convolutional layers, the obtained node v's hidden state $\mathbf{h}_v^k$ contains the information of its k-hop neighbors.

Then, the bi-directional spatial cascade convolutional layers can be represented as follows:

First, each node in cascade c obtains an initial embedding, and the g_c^t can be represented as $\mathbf{H}_c^0 = \mathbf{X}_c$, where $\mathbf{X}_c \in \mathbb{R}^{n \times d}$ is the stacked initial embedding of each node in g_c^t, n is the number of sampled nodes, and d is dimension size of node embedding. Then, the bi-directional spatial convolution of cascade c can be represented as:

$$\mathbf{H}_{N(c\_in)}^k = mean(\mathbf{A}_c^{in} \cdot \mathbf{H}_c^{k-1}) \tag{2}$$

$$\mathbf{H}_{N(c\_out)}^k = mean(\mathbf{A}_c^{out} \cdot \mathbf{H}_c^{k-1}) \tag{3}$$

$$\mathbf{H}_c^k = relu((\mathbf{H}_{N(c\_in)}^k, \mathbf{H}_{N(c\_out)}^k, \mathbf{H}_c^{k-1}) \cdot \mathbf{W}_k + \mathbf{b}_k) \tag{4}$$

where $\mathbf{H}_{N(c\_in)}^k \in \mathbb{R}^{n \times d}$ and $\mathbf{H}_{N(c\_out)}^k \in \mathbb{R}^{n \times d}$ are, respectively, the aggregated information of nodes' neighbors from the information inflow and outflow directions, $\mathbf{W}_k \in \mathbb{R}^{3d \times d}$ and $\mathbf{b}_k \in \mathbb{R}^{n \times d}$ are parameters learned during training, and $\mathbf{H}_c^k \in \mathbb{R}^{n \times d}$ is the stacked hidden states of each node after k layers. In the end, we can obtain the $\mathbf{H}_c^k$ that contains the updated embedding of each node as well as the structural information of the cascade graph.

3.2.4. Temporal Information Aggregation Layer

The time effect has been proven to have a critical effect on the viral spread of online content, where the rich-get-richer effect and the time decay effect are pervasive in cascade evolving patterns [14]. However, the extant graph representation approaches [16,54] focus on the structural information at the path level or graph level, making it difficult to incorporate the temporal information of nodes'/users' retweeting behaviors. In our work, we propose an innovative approach to capture the spreading dynamics by aggregating the information of nodes in the same time window.

Dynamic features extraction: We assume that a dynamic cascade graph $g_c^t = (V_c^t, E_c^t, T_c^t)$ is observed with a time duration t after its origination, and the time label of node v is $\{t_v = t_v^r - t_0^c\}(0 \leqslant t_v \leqslant t, v \in V_c^t)$, where t_v^r is the time when node v retweets the message and t_0^c is the original post time. We divide the time duration t into m time windows. For an arbitrary node in the cascade, which time window it belongs to is decided by the following process:

If $(i-1) \times \frac{t}{m} \leq t_v < i \times \frac{t}{m} (i \in \mathbb{Z}, i \leq m)$: node v belongs to the ith time window.

As shown in Figure 3, according to the nodes in each time window, the matrix $\mathbf{S}_c \in \{0,1\}^{m \times n}$, which contains the spread information of each time window, can be generated. To model the temporal dependence of viral spread, we first aggregate the information of nodes in each time window and then input the aggregated results to a gated recurrent unit (GRU). The process of extracting temporal features of cascade c can be represented as:

$$\mathbf{T}_c = \mathbf{S}_c \cdot \mathbf{H}_c^k \tag{5}$$

where $\mathbf{T}_c \in \mathbb{R}^{m \times d}$ is the stacked initial states of observation time windows. Let $\mathbf{t}_c^i \in \mathbb{R}^d$ denote the initial state of ith time window, which is a row of $\mathbf{T}_c$; the reset gate $\mathbf{r}_c^i \in \mathbb{R}^d$ is computed as:

$$\mathbf{r}_c^i = \sigma((\mathbf{t}_c^i, \mathbf{u}_c^{i-1}) \cdot \mathbf{W}_r) \tag{6}$$

where σ is the sigmoid activation function, $\mathbf{W}_r \in \mathbb{R}^{2d\times d}$ are parameters learned during training, and $\mathbf{u}_c^{i-1}$ is the output state of $(i-1)^{th}$ time window. The update gate $\mathbf{z}_c^i \in \mathbb{R}^d$ is shown as Equation (7), where $\mathbf{W}_z \in \mathbb{R}^{2d\times d}$ are also trainable parameters.

$$\mathbf{z}_c^i = \sigma((\mathbf{t}_c^i, \mathbf{u}_c^{i-1}) \cdot \mathbf{W}_z) \tag{7}$$

After that, the output state of ith time window is computed as:

$$\mathbf{u}_c^i = (1 - \mathbf{z}_c^i) \odot \mathbf{u}_c^{i-1} + \mathbf{z}_c^i \odot \widetilde{\mathbf{u}}_c^i \tag{8}$$

where $\widetilde{\mathbf{u}}_c^i = tanh((\mathbf{t}_c^i, \mathbf{r}_c^i \odot \mathbf{u}_c^{i-1}) \cdot \mathbf{W})$, $\odot$ is the element-wise multiply between vectors, $\mathbf{W} \in \mathbb{R}^{2d\times d}$. The output hidden state of each time window contains information on spreading dynamics as well as the structural information of the diffusion network.

Time decay effect: Since the influence of the message usually declines with time passing, the time decay effect is considered as another important factor in popularity prediction. In our work, due to the observation time t being divided into several time windows, instead of constructing functions to describe the time decay effect of viral spread, our proposed model directly learns the time decay effect of each divided time window. Specifically, a trainable parameter λ_i, which is used to depict the time decay effect of the ith time window, is given. A weighted sum-pooling approach is adopted to aggregate all the output states of m time windows where the final representation $\mathbf{R}_c$ of cascade c that captures both structural and temporal information is obtained.

$$\mathbf{R}_c = \sum_{i=0}^{m} \lambda_i \mathbf{u}_c^i \tag{9}$$

3.2.5. Output Layer

The output layer of our model is a fully connected neural network, taking the learned cascade representation $\mathbf{R}_c$ as input and outputting the final prediction of growth size $MLP(\mathbf{R}_c)$, where MLP stands for a multi-layer perception. In the end, the eventual objective function to be minimized is defined as:

$$O = \frac{1}{|C|} \sum_{c=1}^{|C|} (MLP(\mathbf{R}_c) - \Delta s_c)^2 + 0.001n \tag{10}$$

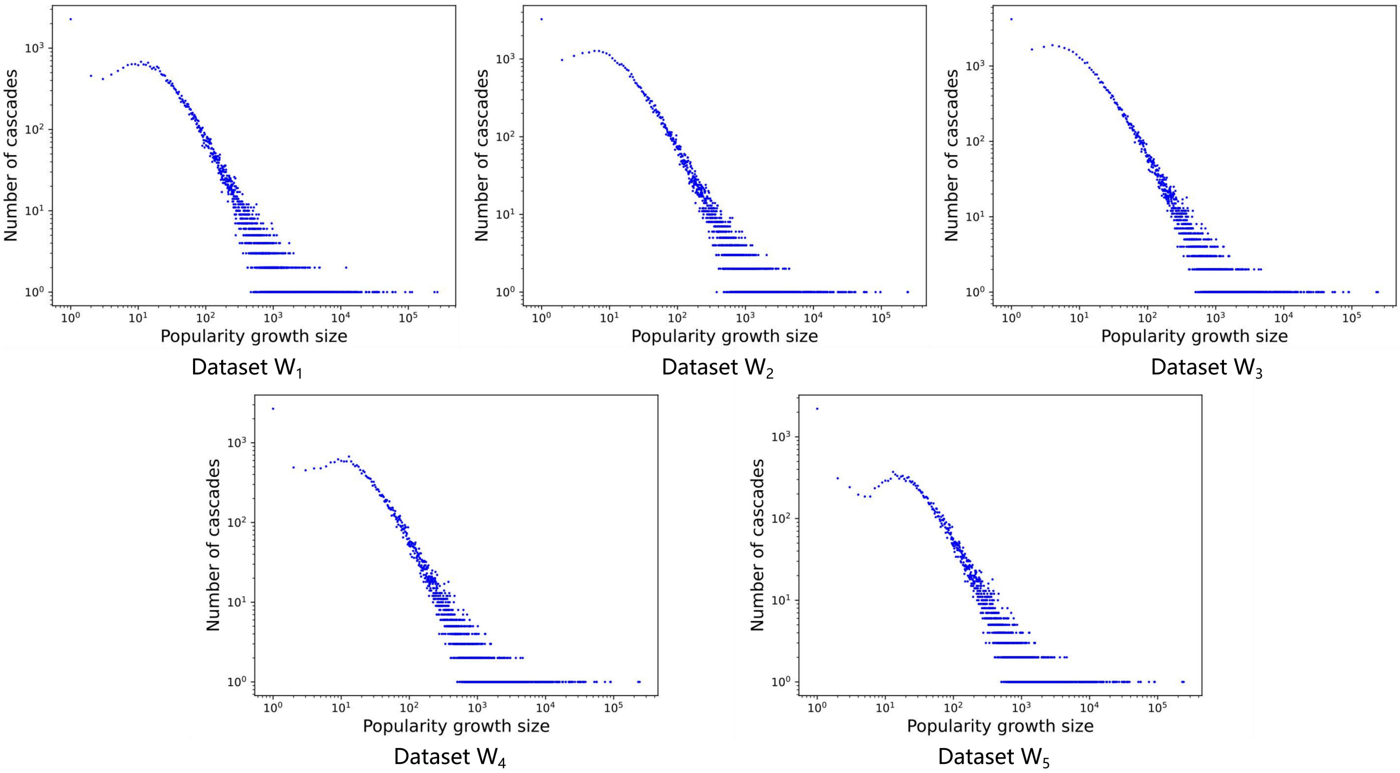
where Δs_c is the cascade c's actual growth size, $|C|$ is the total number of retweet cascades, n is the number of sampled nodes, and 0.001 is the weight which means that it is acceptable if an increase in sampling volume of 100 results in a loss reduction of more than 0.1.

4. Empirical Investigation

In this section, we apply our model to the real viral spread scenario in social media to evaluate the performance of our model. We also compare our model with other state-of-the-art popularity prediction methods to illustrate the advantages of ViralGCN. In addition, we make several variants of our model to test the effectiveness of three main components in our model. Moreover, we adopt T-SNE to visualize the learned representations of nodes as well as cascade graphs to explain what ViralGCN learns from the initial cascade graphs and how they influence future popularity.

4.1. Dataset

The dataset used in our research is the Sina Weibo retweet dataset, which is generated by [17] and is widely used in recent popularity prediction studies [17,18,54] as well. This dataset records the diffusion paths in 24 h of 119,313 original micro-blogs that were posted on 1 June 2016. In addition, the exact timing of each user's retweet is also included in this dataset, allowing our model to capture the dynamic features of spreading. Because the spread dynamics are different between day and night, we follow [17,18] and filter out micro-blogs posted before 6:00 AM or after 18:00. Moreover, according to previous experiments [16–18], the length of the observation time window (i.e., the time duration t after posting) can significantly influence models' performances, and the proportion of small cascades that have little future growth in the dataset may also affect the model's overall prediction errors. Therefore, to comprehensively test the performance of ViralGCN, we construct five sub-datasets based on different observation times and initial cascade sizes. Specifically, the observation time in W_1, W_2, W_3 is, respectively, 1 h, 2 h, and 3 h after posting, and the micro-blogs with an initial cascade size no less than 10 are selected. W_4 and W_5 include the micro-blogs with an initial cascade size of no less than 20 and 30, respectively, and both observation times are 3 h after posting. The statistics are reported in Table 3. Moreover, in this study, the retweet popularity of a micro-blog within 24 h after origination is regarded as its final popularity. Thus, the actual growth size of cascade c is computed as $\Delta s_c = |V_c^{24}| - |V_c^t|$, where t is the observation time. Figure 4 shows the distribution of popularity growth of all five datasets.



Dataset W$_1$

Dataset W$_2$

Dataset W$_3$

Dataset W$_4$

Dataset W$_5$

Figure 4. The distribution of five datasets.

Table 3. Statistics of datasets

Dataset **Observation Time** **The Initial Retweet Popularity**		**W1** **1 h** $N \geq$ **10**	**W2** **2 h** $N \geq$ **10**	**W3** **3 h** $N \geq$ **10**	**W4** **3 h** $N \geq$ **20**	**W5** **3 h** $N \geq$ **30**
Number of micro-blogs	Train	27,487	33,207	36,365	23,148	18,010
	Val	5890	7116	7792	4960	3859
	Test	5890	7116	7793	4960	3859
Avg. number of nodes per graph	Train	115.27	132.24	143.74	214.7	265.36
	Val	118.00	130.95	146.90	217.82	264.13
	Test	125.00	134.94	146.86	224.43	268.28
Avg. number of edges per graph	Train	127.30	148.95	164.28	247.33	307.40
	Val	133.61	152.84	159.80	247.51	310.86
	Test	136.38	153.72	164.88	253.78	313.53

4.2. Evaluation Metric

The mean squared log-transformed error (MSLE), an effective indicator to measure the difference between predicted values and actual values, is adopted as the evaluation metric in this paper. The MSLE is a variant of MSE (mean squared error), and it is frequently applied in regression tasks and popularity prediction studies [16–18]. The definition of MSLE is shown as follows:

$$MSLE = \frac{1}{|C|} \sum_{c=1}^{|C|} SLE_c^2 \tag{11}$$

where $|C|$ is the total number of cascades, an arbitrary cascade c's squared log-transformed error $SLE_c = (\log(\Delta s'_c + 1) - \log(\Delta s_c + 1))^2$, $\Delta s'_c$ is the predicted increment of cascade size, and Δs_c is the actual growth size of the cascade.

4.3. Performance Comparison Experiment

In this experiment, we adopt several common and state-of-the-art cascade prediction approaches as the baselines to compare with our proposed ViralGCN, including the feature-based methods, the DeepCas, the DeepHawkes, the DMT-LIC, and the CasCN.

4.3.1. Baseline Methods

Feature-linear: The linear regression is one of the most common approaches used to model the relationship between online content popularity and the hand-crafted features [11,12,40,41]. In this paper, we extract several frequently used structural and temporal features that can be generalized across all the datasets in this study, including the number of leaf nodes, the average and the max degree of nodes, the average and max path length, the average time that elapsed between the message origination and each retweet, and the average and the maximum time interval between two successive retweets.

Feature-deep: Except for linear regression, we also use a neural network to combine the selected features with the cascade growth size in a non-linear model. We calculate the values of the selected features for each observed cascade graph in our datasets. Then, the obtained features vectors are fed to both the linear regression model and fully connected neural network to estimate the increment of cascade sizes.

DeepCas [16]: It is one of the state-of-the-art graph representation models for popularity prediction, which extracts the structural features of information diffusion networks by using the random walk to sample a series of node sequences from the initial cascade graph.

DeepHawkes [17]: It combines the deep learning method with the generative approach (i.e., the Hawkes process), which makes each information diffusion path an input of a recurrent neural network. This approach mainly considers the temporal information of cascade growth.

DMT-LIC [42]: Another one of the state-of-art graph-based deep learning frameworks for popularity prediction, which uses a deep multi-task learning framework to capture both spatial and temporal dynamics of a cascade.

CasCN [18]: It is a model that predicts cascade growth size through graph convolution in the Fourier domain. It adopts a recurrent neural network to model temporal dynamics.

4.3.2. Experiment Settings

For DeepCas, we set the number of random walk $K = 200$ sequences with walk length $T = 10$. DeepHawkes's hidden layer of each GRU has 32 units. The CasCN's Chebyshev coefficient $K = 2$. For DMT-LIC, the hidden layer of each RNN is 32 units, and the hidden dimensions of the one-layer MLP are 32. In addition, some shared parameter settings are as follows: the dimensionality of node embeddings is 64, the length of the divided time window is 10 min, the dimensionality of nodes' hidden states is 32, the batch size is 20, the hidden dimensions of the two-layer MLP are 32 and 16, and the learning rate is 1×10^{-3}. For our proposed ViralGCN, the hidden layer of each GRU is 32 units, and the number of spatial cascade convolution layers is $k = 3$. The hyper-parameter n ranges from 100 to 1000 taking intervals of 100, and the training process will stop when the model converges.

4.3.3. Results

We apply our proposed model and seven baseline methods on five datasets generated from real viral spread events on Weibo. In this section, we are going to compare the performances of all models. The result statistics are shown in Table 4. The differences between the five datasets are the observation time and initial sizes of selected cascades, by which we can better investigate the models' prediction abilities.

It can be clearly seen from the results of W_1, W_2, and W_3 that the overall performances of all models upgrade with the extending observation time window, suggesting that longer observation brings more predictive information. In results of W_3, W_4 and W_5 where the observation time is fixed, the performances of all of the models degrade with the growing initial sizes of selected cascades. The results show that the balance of the dataset affects the models' overall performances, and accurate predictions for cascades with large initial sizes are much more difficult than for small ones. To better illustrate the effect of observation time and dataset balance, we provide the validation loss of the proposed ViralGCN on five datasets with training epochs in Figure 5.

Feature-based vs. graph-based. Compared to the other graph-based models, two feature-based models, i.e., the feature-linear and the feature-deep, achieve relatively high prediction errors for all of the five datasets. For instance, when the observation time was 1 h and the minimum initial popularity is set to 10, the feature-linear model obtained the highest prediction error with $MSLE = 3.768$ and the feature-deep model obtained the second highest prediction error with $MSLE = 3.523$. The results indicate that some predictive information is excluded by the hand-crafted features, and the performance of the feature-based model significantly depends on the quality of adopted features. In addition, the better performances of the deep learning model over the linear model show that possible non-linear relationships exist between adopted features and future popularity growth.

Structural information only vs. a combination of temporal information. We compare the performances of the graph-based models that only consider cascades' structural information (i.e., DeepCas) with the models that combine both structural and temporal information (i.e., DeepHawkes, DMT-LIC, CasCN, and the proposed ViralGCN). It is clear from Table 4 that the prediction error for DeepCas on the five data sets is 2.922, 2.694, 2.603, 2.710, and 2.847, which is significantly higher than DeepHawk, DMT-LIC, and CasCN, respectively. The results strongly show that the overall prediction errors can be decreased by incorporating temporal information into the model.

ViralGCN vs. other graph-based methods. Finally, our proposed model performs significantly better than not only the feature-based models but also mainstream graph-based models for all five datasets with varying observation time windows and initial sizes of selected cascades. Specifically, ViralGCN achieves excellent prediction results with $MSLE$s of 2.068, 1.460, 1.206, 1.423, and 1.527 on the five datasets. Compared with the current state-of-the-art method CasCN, the errors are reduced by 10.7%, 31.9%, 37.0%, 29.1%, and 26.7%, respectively. Overall, our proposed ViralGCN model shows a strong ability to predict the popularity of online viral content.

Table 4. Comparison of performances.

Data Set **Observation Time** **The Initial Retweet Popularity**	W_1 **1 h** $N \geq$ **10**	W_2 **2 h** $N \geq$ **10**	W_3 **3 h** $N \geq$ **10**	W_4 **3 h** $N \geq$ **20**	W_5 **3 h** $N \geq$ **30**
Feature-linear	3.768	3.594	3.267	3.383	3.496
Feature-deep	3.523	3.440	3.105	3.287	3.331
DeepCas	2.922	2.694	2.603	2.710	2.847
DeepHawk	2.430	2.202	2.168	2.251	2.337
DMT-LIC	2.474	2.310	2.129	2.183	2.297
CasCN	2.317	2.146	1.915	2.008	2.082
ViralGCN (Proposed)	2.068 *	1.460 *	1.206 *	1.423 *	1.527 *

* means the result is significantly different from the extant models at 0.01 level.

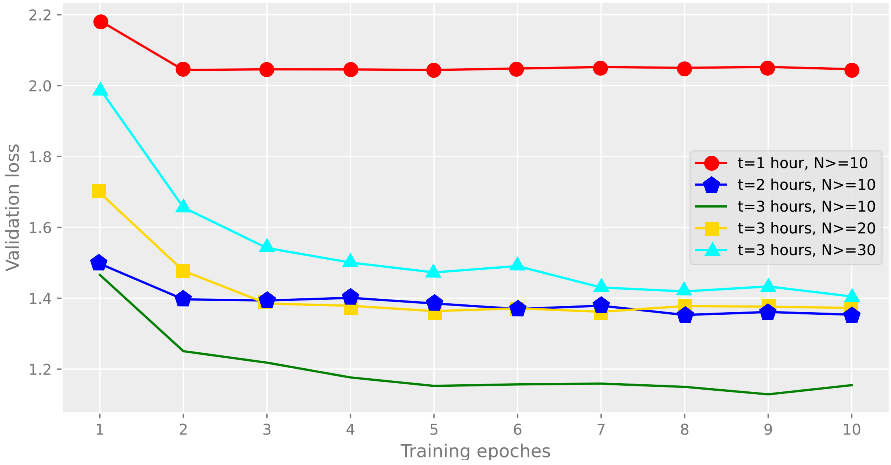


Figure 5. Loss of ViralGCN on the validation set of all five datasets.

4.4. Ablation Experiment

The main objective of this experiment is to evaluate the effectiveness of components in our proposed model so that we make several variants of ViralGCN and compare the performances of these modified models on the above five datasets.

4.4.1. The Variants of ViralGCN

ViralGCN-fixed: The first component of ViralGCN is the node-sampling process that is developed to handle oversize cascade graphs and to improve the computation efficiency. To test if our node-sampling process harms the model's performance, we create *ViralGCN-fixed*, the input size of which is set to fix 1000.

ViralGCN-max: The aggregator function is one of the most important components in the spatial cascade convolutional layer. The original ViralGCN adopts the mean aggregator to assemble information about a node's neighbors. Due to the max aggregator being also widely used in the graph learning domain, we build *ViralGCN-max*, which uses the max aggregator to test the effectiveness of ViralGCN's mean aggregator by comparing their performances.

ViralGCN-undirected: To test the effectiveness of our proposed bi-directional spatial graph convolutional method, which extracts features of both information inflow and outflow directions, we make *ViralGCN-undirected*, which treats cascades as undirected graphs and does not distinguish a node's neighbors from different directions.

ViralGCN(no time effect): It is a variant of ViralGCN without considering any spreading dynamics or time decay effects, which makes predictions only based on the representations obtained from the spatial convolutional layer. This model is constructed to test if it is necessary to consider the temporal information in popularity prediction.

4.4.2. Results

The statistical results of all modified models are shown in Table 5. We can see that there is no significant difference between the performances of ViralGCN-max and the original ViralGCN, demonstrating that both max-pooling and mean-pooling are effective aggregator functions for ViralGCN to extract nodes' local structural features.

Additionally, compared with ViralGCN-undirected, the original ViarlGCN with a bi-directional convolution obtains remarkably better overall performances for all datasets. The results prove that information directions can provide more useful information about cascade structure, and it is better to distinguish different information diffusion directions in cascade prediction.

Moreover, it can also be seen that omitting the time effect leads to a significant increase in prediction errors where the ViralGCN (no time effect) does not perform as well as the original ViralGCN on all data sets. It suggests that temporal information is as crucial as cascade structures to predict the popularity of viral spread.

Moreover, it can be found that compared to the original ViralGCN, ViralGCN-fixed's MSLE in all five datasets is, respectively, reduced by 1.69%, 0.062%, 3.07%, 3.87%, and 4.65%, but the average consuming time of one epoch training is, respectively, increased by 549%, 317%, 190%, 160%, and 132%. The results clearly show that the proposed node-sampling process is effective to extract the main information of the oversize cascade without inputting the whole graph.

In summary, the bi-directional spatial convolution layers and the temporal information aggregation layer of our proposed ViralGCN are effective to extract valuable information from cascades. Moreover, both structural features of the cascade graph and temporal dynamics of viral spread play important roles in reducing errors of popularity prediction. The node sampling method can greatly enhance the computation efficiency by sacrificing a little prediction accuracy. The experimental results demonstrate the effectiveness and necessity of all three components in the ViralGCN model.

Table 5. Comparison of ViralGCN and its variants.

Data Set	W_1	W_2	W_3	W_4	W_5
Observation Time	**1 h**	**2 h**	**3 h**	**3 h**	**3 h**
The Initial Retweet Popularity	$N \geq 10$	$N \geq 10$	$N \geq 10$	$N \geq 20$	$N \geq 30$
ViralGCN-max	2.063	1.468	1.195	1.423	1.521
ViralGCN-undirected	2.195 *	1.603 *	1.525 *	1.710 *	1.907 *
ViralGCN-no time effect	2.594 *	2.512 *	2.463 *	2.553 *	2.680 *
ViralGCN-fixed	2.033	1.451	1.169 *	1.368 *	1.456 *
ViralGCN (original)	2.068	1.460	1.206	1.423	1.527
Best sampling volume of ViralGCN	300	400	500	500	500
Avg. 1 epoch training time of ViralGCN	204s	415s	780s	708s	651s
Avg. 1 epoch training time of ViralGCN-fixed	1324s	1732s	2269s	1842s	1511s

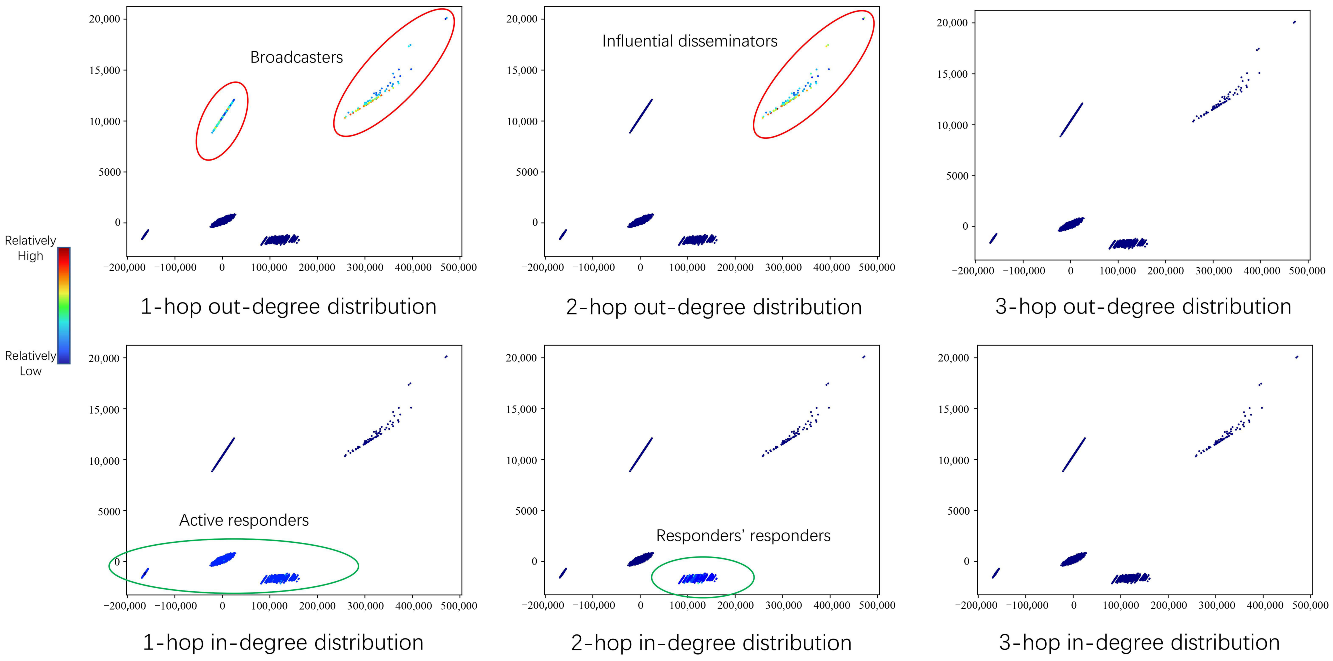
* means the result is significantly different from the original ViralGCN at 0.01 level.

4.5. Study of the Learned Features

In this section, we are going to provide some insights into actual viral spread mechanisms from the perspective of artificial intelligence. Specifically, we try to use the T-SNE (i.e., t-distributed stochastic neighbor embedding) to project the output high-dimensional nodes' embeddings and cascades' embeddings onto a two-dimensional plane, combining with the visualization method to investigate which features are learned by our proposed spatial-temporal cascade convolutional framework and how they affect the future popularity growth of viral online content. Although ViralGCN does not achieve the best overall performance on dataset W_5, it learns more valuable relationships between the features of early viral spread and its future popularity growth. Hence, for presentation convenience, only the visualizations of dataset W_5's test results are given. The visualization results of the other datasets are provided in the Appendix A.

4.5.1. The Learned Users' Features

We first take the T-SNE to project 789854 nodes' embeddings, which are output by the bi-directional spatial convolutional layers onto a two-dimensional plane, as shown in Figure 6. The T-SNE (i.e., t-distributed stochastic neighbor embedding) [55] is a widely used method for visualizing high-dimensional data, by which similar objects are modeled into the same cluster, while dissimilar objects are vice versa. It can be seen from Figure 6 that all the nodes are projected into five different clusters. To further investigate the differences between nodes in distinct clusters, we, respectively, color every node according to its 1, 2, and 3-hop in-degrees and out-degrees as the feature of a node in the graph is mainly determined by its neighbors. After that, the results in Figure 6 show that four main types of users that play different roles in the viral spread are identified by ViralGCN. Specifically, the first type of users that only have relatively high 1-hop out-degrees are projected in the top left cluster. We call these users broadcasters; they are usually influential in their communities and can easily bring retweets from their friends but are less effective in forming a long diffusion path. The second type of users whose 1-hop and 2-hop out-degrees are both relatively high is projected into the top right cluster. We call these users influential disseminators; they can bring retweets not only from their friends but their friends' friends as well (i.e., longer diffusion paths). Both broadcasters and influential disseminators are usually root posters in the viral spread. Three clusters of nodes with relatively high 1-hop in-degrees are projected to the bottom left of the figures. These users are called active responders; they frequently retweet messages from others but are less influential at bringing new retweets. Note that the right cluster of active responders also have relatively high 2-hop in-degrees, which means that they not only retweet messages from those influential users but also other active responders. We call these users the responders' responders; they can be easily influenced by others and usually play the role of followers. Moreover, almost all nodes have low 3-hop out-degrees and 3-hop in-degrees, implying that the length of the most retweet paths at an early stage ($t = 3$ h in dataset W_5) is within three users.



1-hop out-degree distribution

2-hop out-degree distribution

3-hop out-degree distribution

1-hop in-degree distribution

2-hop in-degree distribution

3-hop in-degree distribution

Figure 6. The projections of learned nodes' representations.

4.5.2. The Learned Cascades' Features

After identifying two important types of users (i.e., broadcasters and influential disseminators) that are capable of influencing others' retweet behaviors, the next objective is to find out which ViralGCN learns about the effects of these users on making online content go viral. Hence, we project the output representation of each initial cascade graph $\mathbf{R}_c$ onto a two-dimensional plane and, respectively, color each cascade by its max. 1-hop out-degree, max. 2-hop out-degree, and actual popularity growth, as shown in Figure 7. The figure on the top left is the distribution of cascades' max. 1-hop out-degrees, from which it can be found that broadcasters generally exist in the initial cascades. Additionally, compared with the distribution of actual popularity growth on the bottom left, there is a clear overlap between the cascade with a strong broadcaster and the cascade with a relatively high growth size. In contrast, it can be seen from the top right figure that the distribution of the initial cascades' max. 2-hop out-degrees is extremely imbalanced, whereas a few cascades with influential disseminators are mainly concentrated in orange circles. Given that it is difficult to illustrate the effects of influential disseminators, we try to investigate the relationship between the popularity growth and the cascade's structural virality by calculating the Wiener Index. The Wiener Index is the mean value of the lengths of the shortest paths between all pairs of nodes [2], where the longer retweet paths generated from those influential disseminators can lead to a larger Wiener Index. We color the projected cascade representations by its Wiener Index in the bottom right figure. Compared with the distribution of cascades' actual popularity growth, it is apparent that the cascade with a relatively high Wiener Index is also likely to achieve relatively large popularity growth in the future. In summary, our proposed ViralGCN is effective at extracting not only nodes' features but the structural features of diffusion networks as well. Moreover, the visualization of cascades' representations suggests that both broadcast and structural virality have a positive effect on future popularity growth.

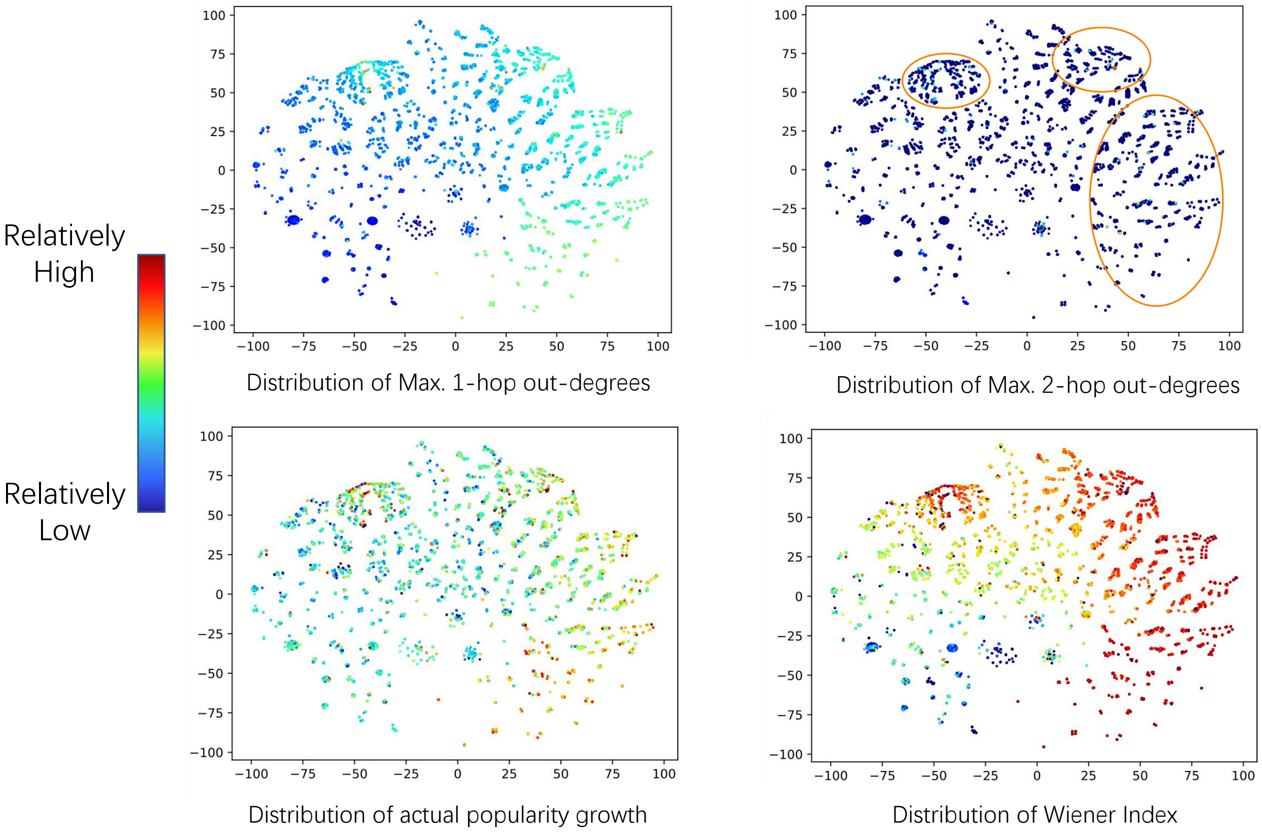
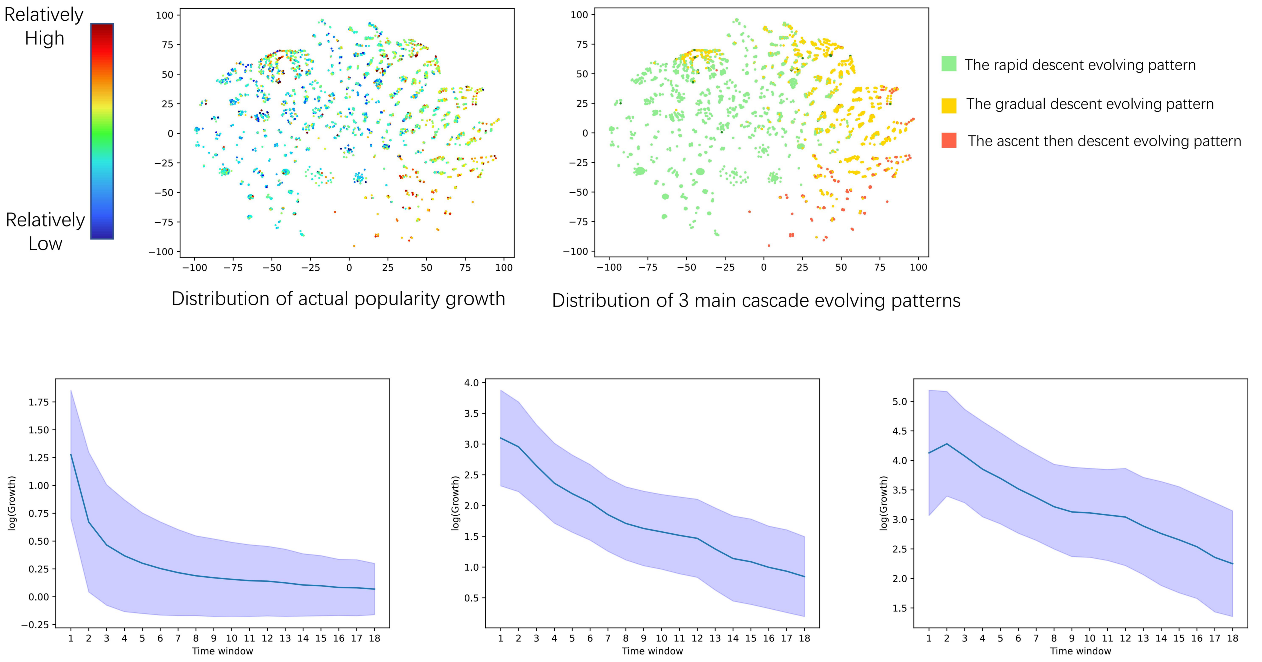


Figure 7. The projections of learned cascades' representations.

4.5.3. The Learned Spreading Dynamics

Except for spatial cascade convolutional layers, another core component of the proposed ViralGCN is the temporal information aggregation layer, which is used to extract the dynamic features of online viral spread. To explain what ViralGCN learns through the temporal information aggregation layer and how the learned temporal features affect the popularity growth, we first try to figure out what the different types of evolving patterns are in the early stage of online viral spread. Hence, we adopt the K-means method to cluster the 3859 micro-blogs in the test dataset of W_5 based on the spreading dynamics during the observation time window, i.e., $t = 3$ h after origination. As a result, three early evolving patterns of viral spread are identified from the test dataset of W_5, as shown in the bottom of Figure 8, where the x-axis is the divided time windows and the y-axis is the log of retweet popularity growth, the blue line is the mean value, and the top and bottom shades are pluses or minus one standard deviation, respectively. We call the left one the rapid descent evolving pattern, whose retweet popularity is relatively lower than the other two patterns initially and declines rapidly over time. The middle one is called the gradual descent evolving pattern, which makes relatively higher popularity initially than the rapid descent pattern and declines gradually over time. The last one is called the ascent-then-descent evolving pattern, from which not only a relatively high initial popularity but also a remarkable rise at the second time window can be seen. Additionally, the retweet popularity of the ascent-then-descent evolving pattern declines gradually as well. After this, we color the projected cascades by their early evolving pattern in the top right of Figure 8 to investigate the effects of different early evolving patterns on future popularity growth. Compared with the distribution of actual popularity growth, it is obvious that the cascades with the gradual descent or ascent-then-descent evolving patterns are more likely to gain relatively larger popularity growth in the future.



Distribution of actual popularity growth

Distribution of 3 main cascade evolving patterns

The rapid descent evolving pattern

The gradual descent evolving pattern

The ascent-then-descent evolving pattern

Figure 8. The learned cascade evolving patterns.

4.5.4. The Learned Time Decay Effects

In the temporal information aggregation layer, ViralGCN adopts several trainable parameters λ_i to learn the time decay effect of each time window. To investigate whether the time decay effects can be learned by ViralGCN and how the time decay effects influence popularity prediction, we plot the learned λ_i of each time window as a bar chart in Figure 9. According to the absolute value of the learned λ_i, it can be seen that the height of bars increases with the time window number, suggesting that the more recent retweets are more valuable for predictions. In other words, the proposed ViralGCN is capable of learning the time decay effects of online viral spread. In addition, we also find that some of the learned λ_i is less than 0. The results explain that the timings of retweets may have different effects on future popularity growth. Since the length of the time window is 10 min, the results of learned λ_i imply that the retweets within 20–40 min and 70–80 min after posing, as well as the last 1 h before observation time, can make positive contributions for future popularity growth.

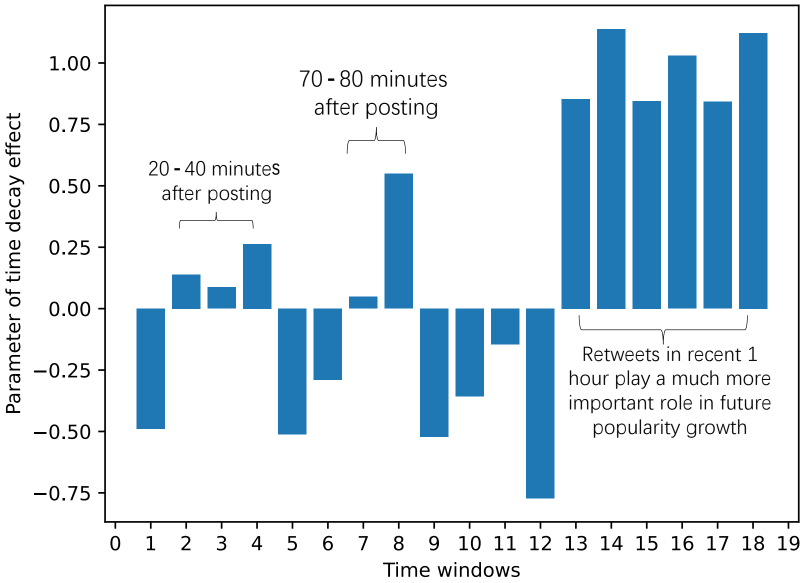


Figure 9. The learned time decay effects.

5. Conclusions

In this paper, we propose an innovative temporal-spatial cascade convolutional framework to predict the future popularity of viral online content that has not been seen before. Our research suggests that the proposed method could provide the following new capabilities to complement existing graph-based popularity prediction methods. First, compared to extant methods, the proposed ViralGCN adopting the adaptive node-sampling process is particularly efficient for handling cascade graphs with large scales. Our ablation experiment shows that the original ViralGCN using the sampling process has more than 100% improvement in computational efficiency over the modified model with expanded input size. Second, we demonstrate that the bi-directional spatial convolution provides a possible solution for the problem of model interpretability, which is a long-standing challenge in graph-based methods. The visualization of node representations allows us to investigate different roles that users play in the viral spread of online content and their effects on future popularity growth. Last, we develop an effective temporal convolution method combined with the time decay effects to fully capture the dynamic features of online viral spread. The results of the ablation experiment show that the model performance can be reduced by up to 73% without considering temporal information.

Another contribution of this paper is that we provide some insights into viral spread mechanisms from the perspective of artificial intelligence. ViralGCN successfully identifies four types of users and three evolving patterns of online viral spread. The visualization results show that both broadcast and structural virality have a positive relationship with future popularity growth. In addition, the cascades with the gradual descent or the ascent-then-descent evolving patterns are more likely to gain large popularity growth in the future, implying that maintaining high popularity for a long duration is one of the keys to making online content go viral. Moreover, our results also indicate that the timing of users getting involved in the cascade could have different effects on its final popularity.

With regard to extensions, our future research may further explore the application of ViralGCN in different online viral spread scenarios. To further validate the ViralGCN model performance and the generalizability of the learned viral spread mechanisms, we consider using more richly sourced data (videos on Youtube, news on Twitter, etc.) in the future. In terms of the ViralGCN framework, figuring out how to better integrate ViralGCN with existing information cascade models is another valuable problem to be studied in the future. A recent study [54] has shown that combining the classical information cascade model with the graph-learning method may help to explain the actual viral spread mechanisms. Such endeavors are fruitful areas for future research.

Author Contributions: Conceptualization, Z.X. and M.Q.; methodology, software, validation, and writing—original draft, Z.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Key Projects of National Social Science Foundation: Innovation and Application of Information Analysis Methods Based on Digital Intelligence Fusion, grant number 22AZD158; Research Start-up Fund of Capital University of Economics and Business, grant number XRZ2022024.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable

Data Availability Statement: The data used in this study are available at https://github.com/CaoQi92/DeepHawkes, accessed on 10 July 2020.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Appendix A

In this appendix, we are going to report the visualization results of the other test datasets as what we have done in Section 4. It should be noted that the visualization results of W_1's test set are less informative since the observation time within 1 h provides little information for ViralGCN to learn. As a result, we only provide the projected cascade's representations of W_1's test set where a lot of cascades are clustered at several points, showing that there is not sufficient information to distinguish the cascades with different popularity growth.

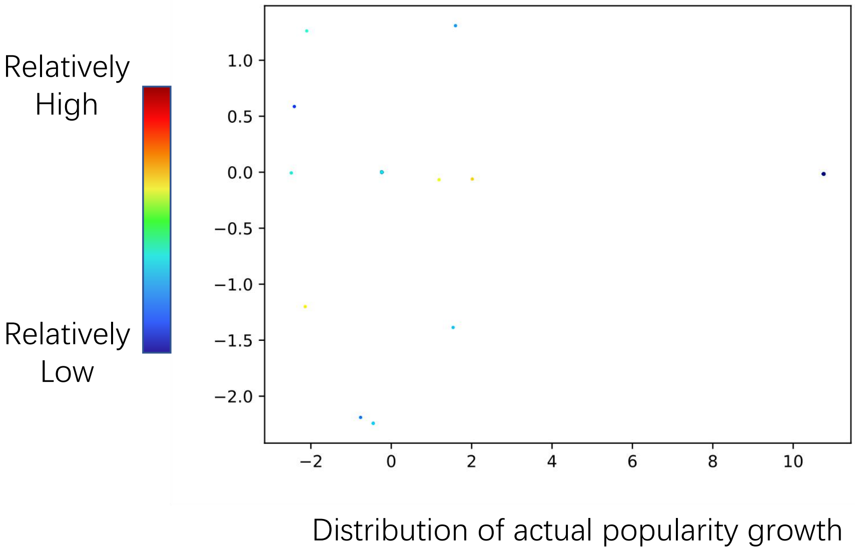
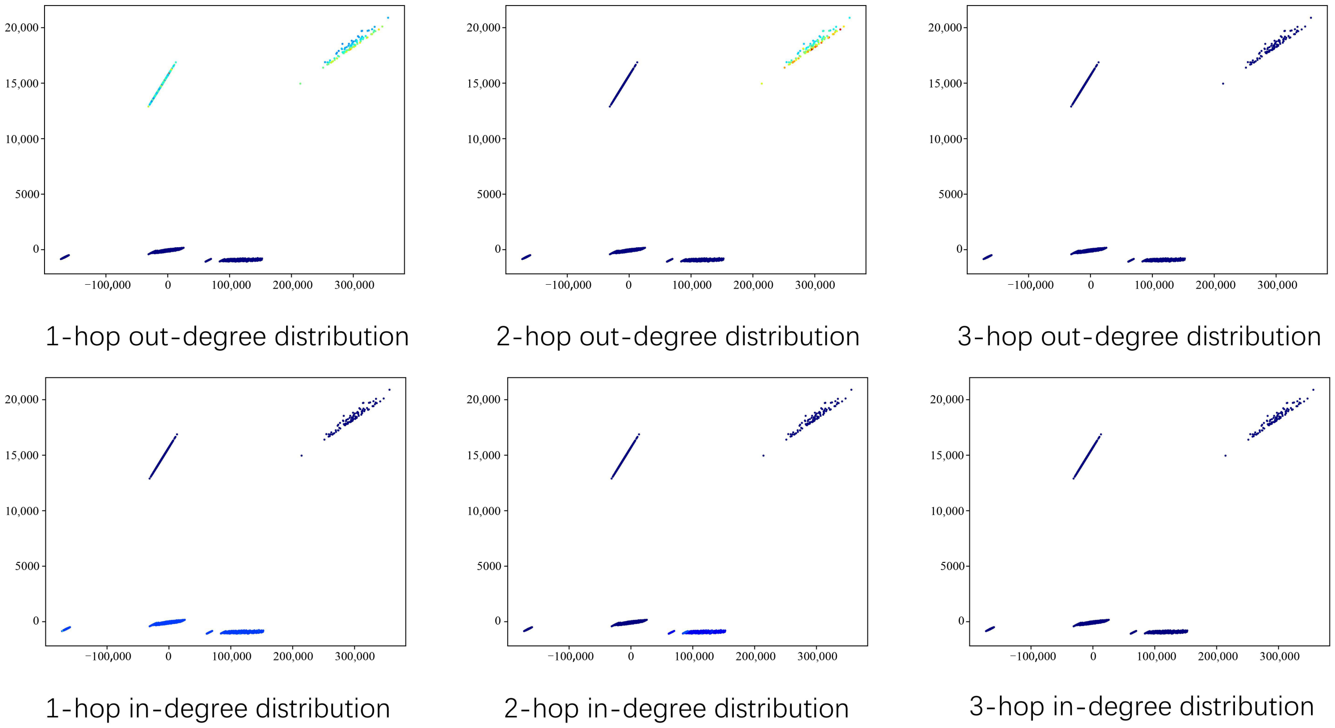


Figure A1. The projected cascade's representations of W_1 test set.

The visualization results of W_2's test set are shown below:



1-hop out-degree distribution

2-hop out-degree distribution

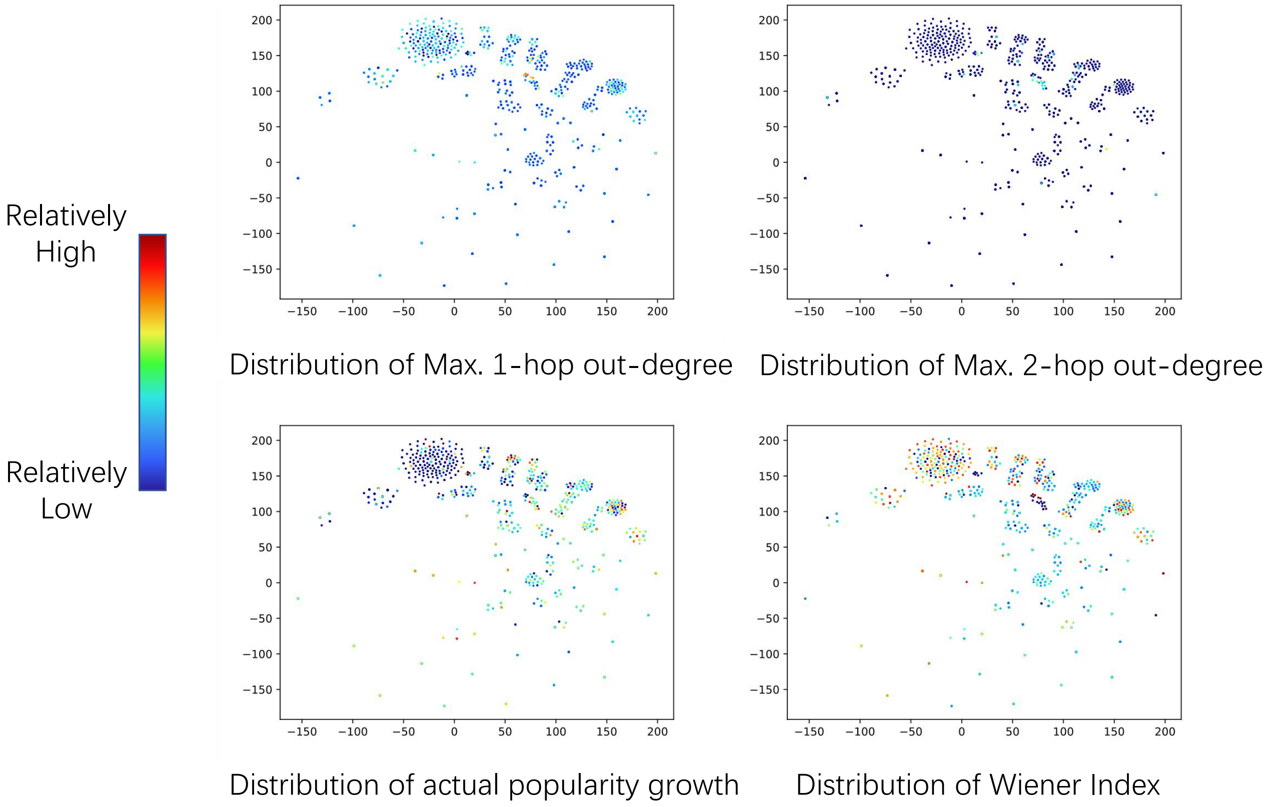
3-hop out-degree distribution

1-hop in-degree distribution

2-hop in-degree distribution

3-hop in-degree distribution

Figure A2. The projections of learned node's representations of W_2 test set.



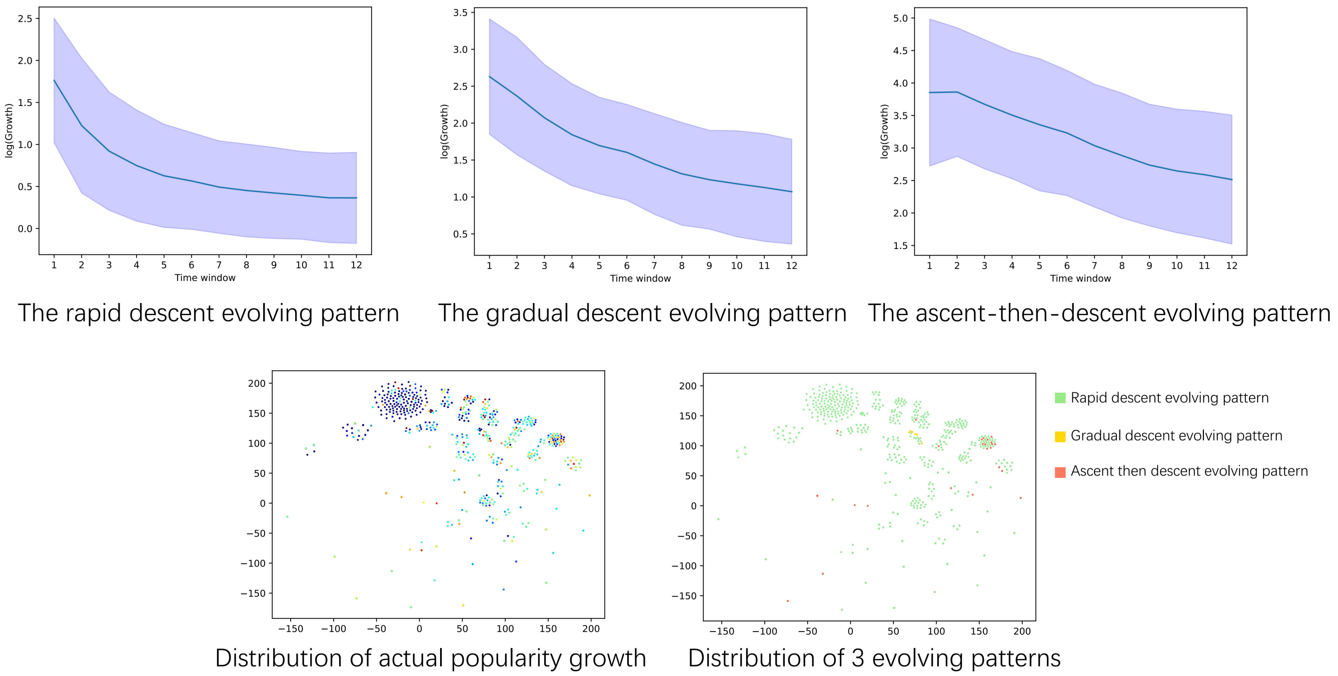
Distribution of Max. 1-hop out-degree

Distribution of Max. 2-hop out-degree

Distribution of actual popularity growth

Distribution of Wiener Index

Figure A3. The projections of learned cascade's representations of W_2 test set.

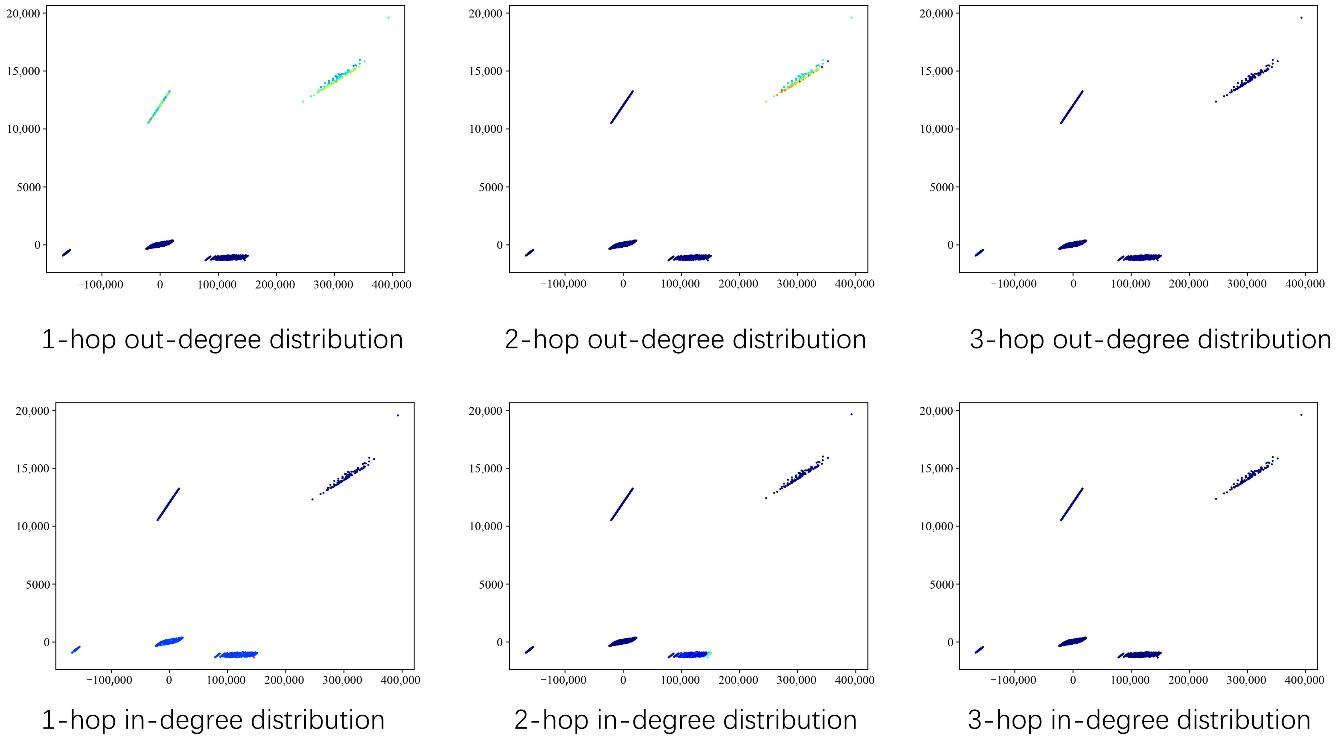


The rapid descent evolving pattern The gradual descent evolving pattern The ascent-then-descent evolving pattern

Distribution of actual popularity growth Distribution of 3 evolving patterns

Figure A4. The learned cascade's evolving patterns of W_2 test set.

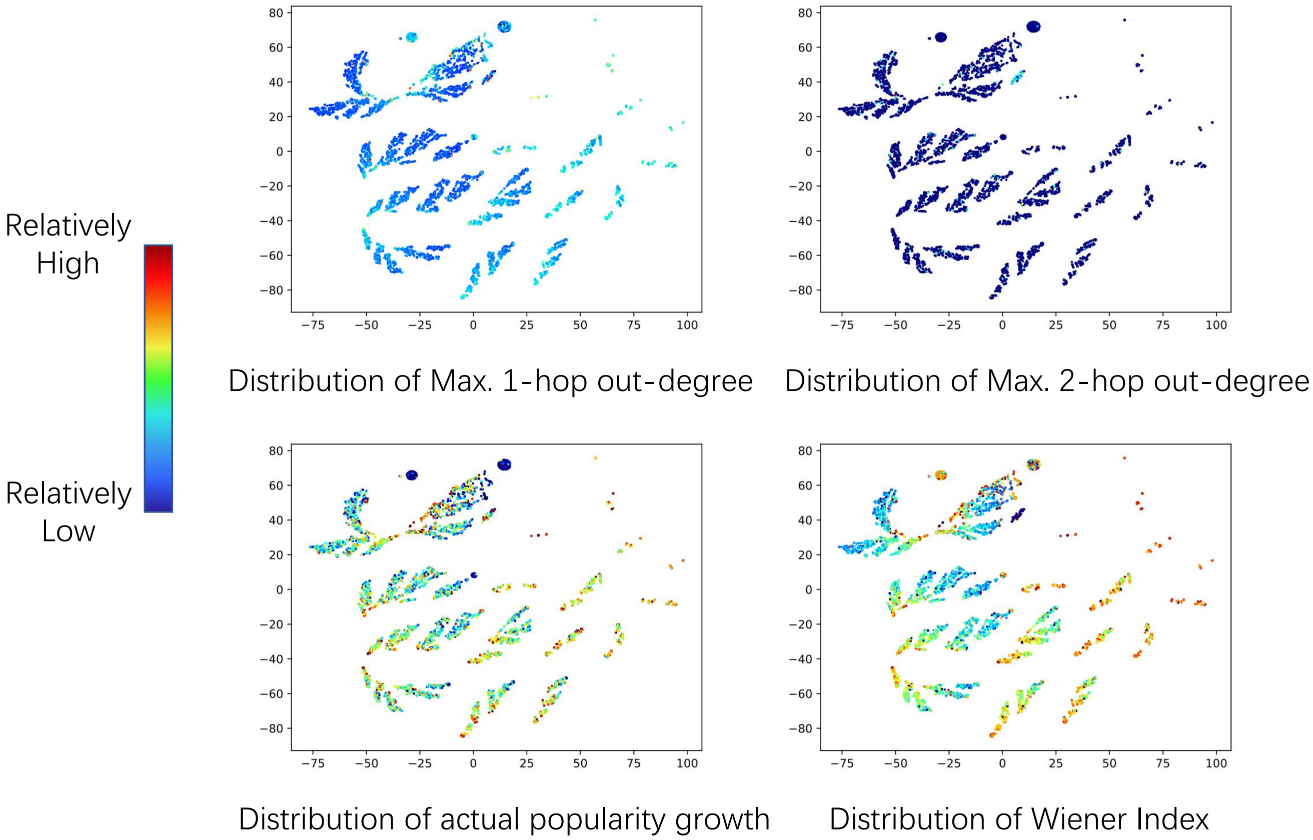
The visualization results of W_3's test set are shown below:



1-hop out-degree distribution 2-hop out-degree distribution 3-hop out-degree distribution

1-hop in-degree distribution 2-hop in-degree distribution 3-hop in-degree distribution

Figure A5. The projections of learned node's representations of W_3 test set.



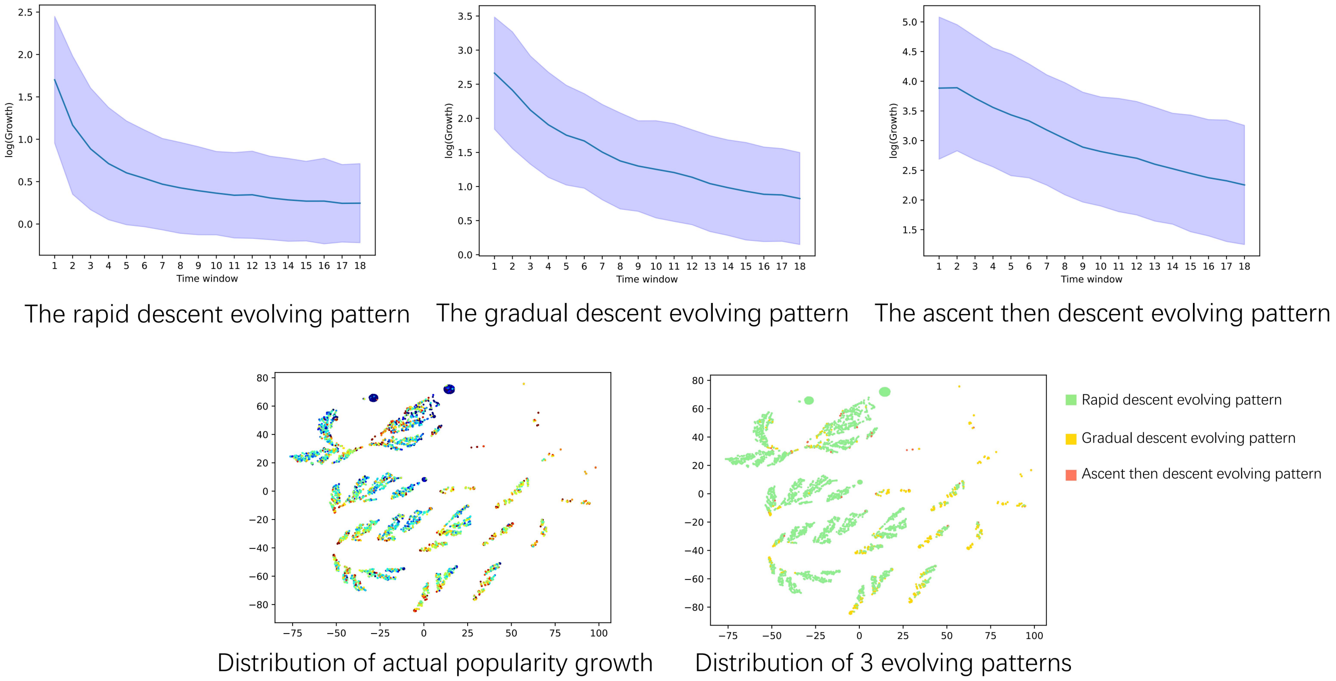
Distribution of Max. 1-hop out-degree

Distribution of Max. 2-hop out-degree

Distribution of actual popularity growth

Distribution of Wiener Index

Figure A6. The projections of learned cascade's representations of W_3 test set.



The rapid descent evolving pattern

The gradual descent evolving pattern

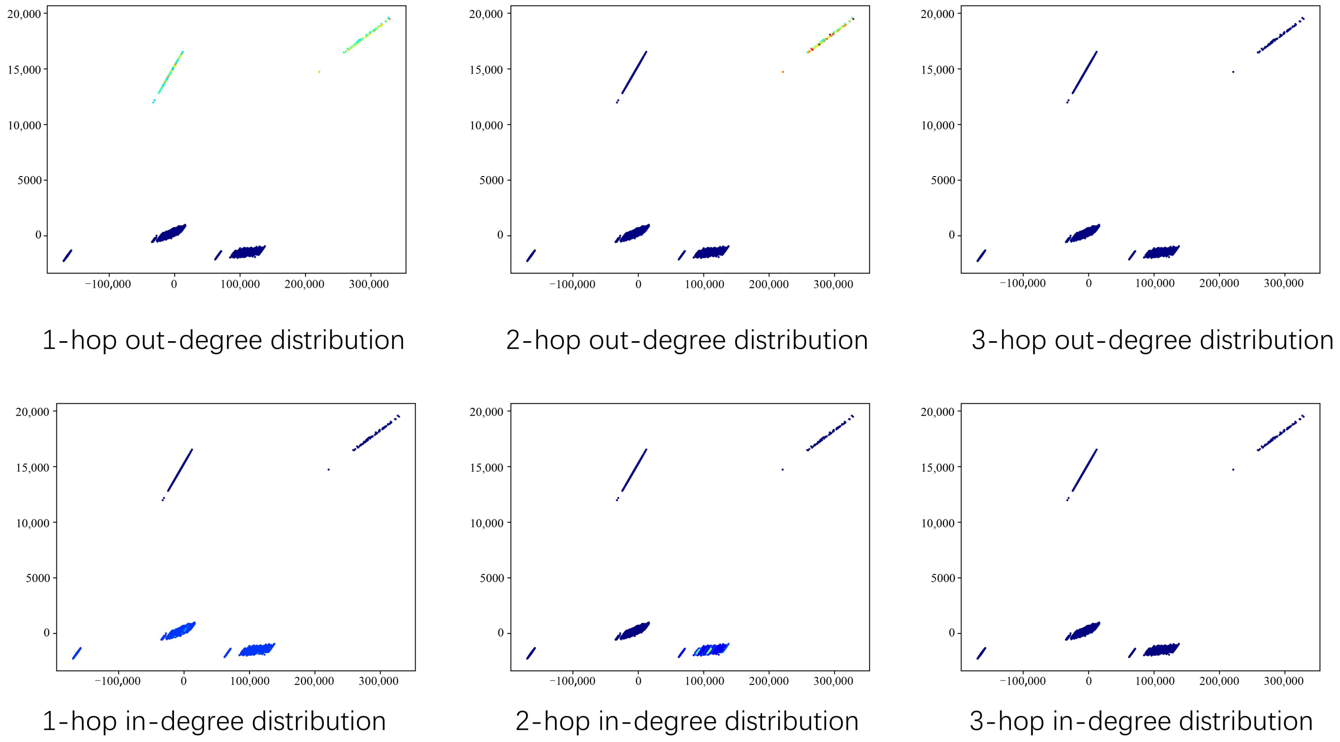
The ascent then descent evolving pattern

Distribution of actual popularity growth

Distribution of 3 evolving patterns

Figure A7. The learned cascade's evolving patterns of W_3 test set.

The visualization results of W_4's test set are shown below:



1-hop out-degree distribution

2-hop out-degree distribution

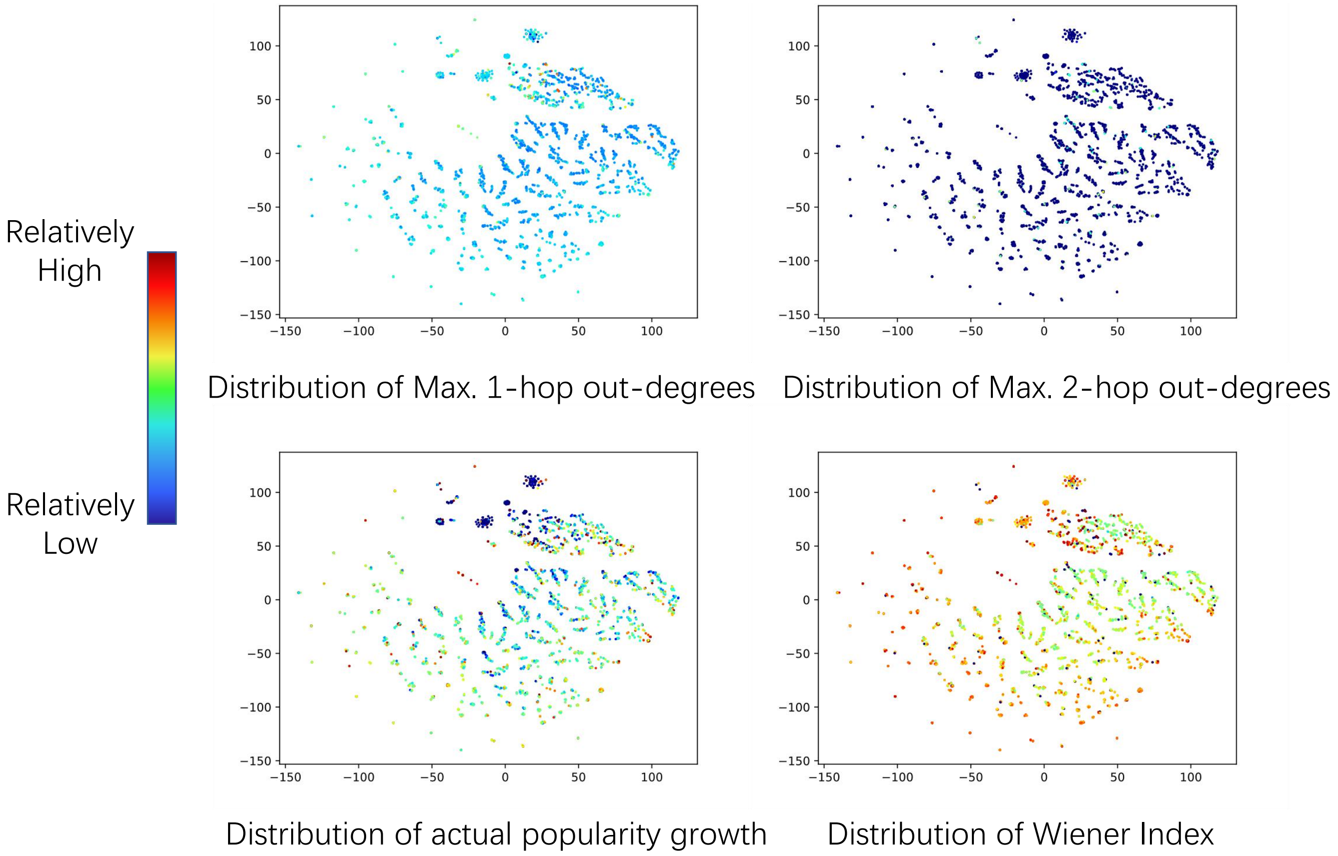
3-hop out-degree distribution

1-hop in-degree distribution

2-hop in-degree distribution

3-hop in-degree distribution

Figure A8. The projections of learned node's representations of W_4 test set.



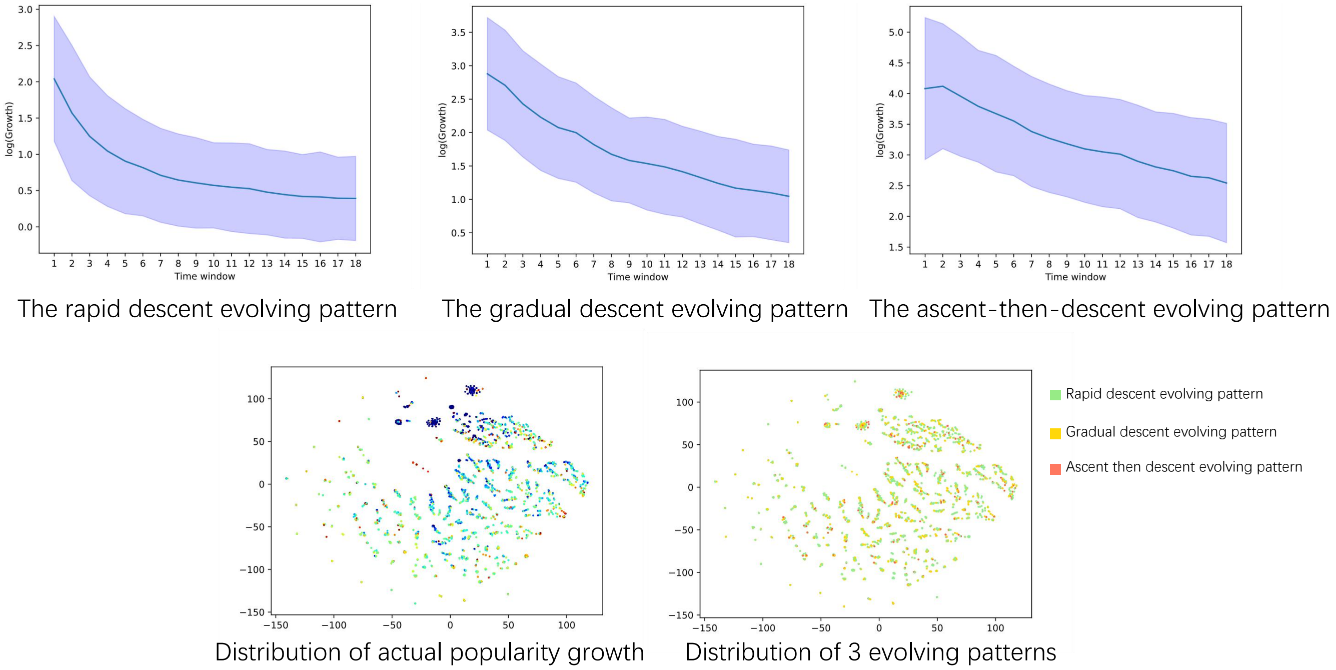
Distribution of Max. 1-hop out-degrees

Distribution of Max. 2-hop out-degrees

Distribution of actual popularity growth

Distribution of Wiener Index

Figure A9. The projections of learned cascade's representations of W_4 test set.

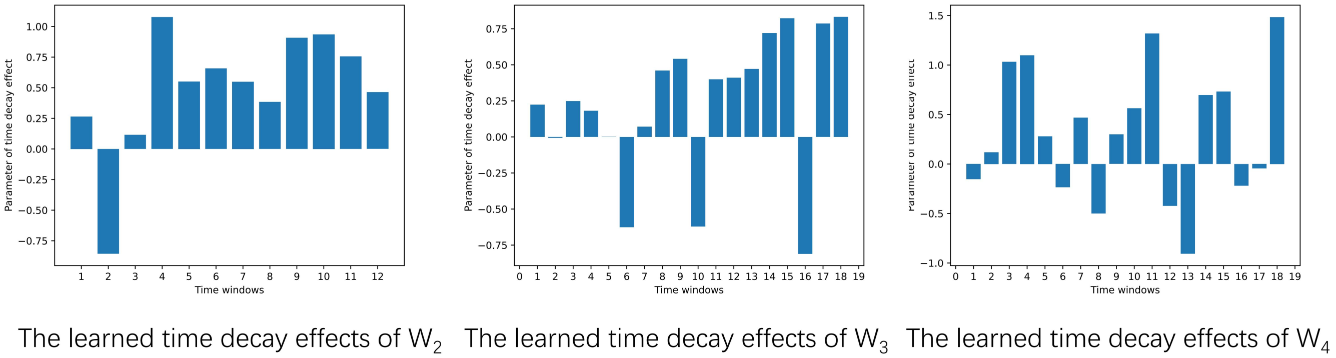


The rapid descent evolving pattern The gradual descent evolving pattern The ascent-then-descent evolving pattern

Distribution of actual popularity growth Distribution of 3 evolving patterns

Figure A10. The learned cascade's evolving patterns of W_4 test set.

The learned time decay effects of W_2's, W_3's and W_4's test sets are shown below:



The learned time decay effects of W_2 The learned time decay effects of W_3 The learned time decay effects of W_4

Figure A11. The learned time decay effects of W_2, W_3 and W_4 test sets.

References

1. Kozinets, R.V.; De Valck, K.; Wojnicki, A.C.; Wilner, S.J. Networked narratives: Understanding word-of-mouth marketing in online communities. *J. Mark.* **2010**, *74*, 71–89. [CrossRef]
2. Goel, S.; Anderson, A.; Hofman, J.; Watts, D.J. The structural virality of online diffusion. *Manag. Sci.* **2016**, *62*, 180–196. [CrossRef]
3. Leonhardt, J.M. Tweets, hashtags and virality: Marketing the Affordable Care Act in social media. *J. Direct Data Digit. Mark. Pract.* **2015**, *16*, 172–180. [CrossRef]
4. Herhausen, D.; Ludwig, S.; Grewal, D.; Wulf, J.; Schoegel, M. Detecting, preventing, and mitigating online firestorms in brand communities. *J. Mark.* **2019**, *83*, 1–21. [CrossRef]
5. Wu, P.C.; Wang, Y.C. The influences of electronic word-of-mouth message appeal and message source credibility on brand attitude. *Asia Pac. J. Mark. Logist.* **2011**, *23*, 448–472. [CrossRef]
6. Babić Rosario, A.; Sotgiu, F.; De Valck, K.; Bijmolt, T.H. The effect of electronic word of mouth on sales: A meta-analytic review of platform, product, and metric factors. *J. Mark. Res.* **2016**, *53*, 297–318. [CrossRef]
7. Cheung, M.; She, J.; Junus, A.; Cao, L. Prediction of virality timing using cascades in social media. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **2016**, *13*, 1–23. [CrossRef]
8. Motoki, K.; Suzuki, S.; Kawashima, R.; Sugiura, M. A combination of self-reported data and social-related neural measures forecasts viral marketing success on social media. *J. Interact. Mark.* **2020**, *52*, 99–117. [CrossRef]
9. Zhou, F.; Xu, X.; Trajcevski, G.; Zhang, K. A survey of information cascade analysis: Models, predictions, and recent advances. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–36. [CrossRef]

10. Gupta, M.; Gao, J.; Zhai, C.; Han, J. Predicting future popularity trend of events in microblogging platforms. *Proc. Am. Soc. Inf. Sci. Technol.* **2012**, *49*, 1–10. [CrossRef]
11. Bandari, R.; Asur, S.; Huberman, B. The pulse of news in social media: Forecasting popularity. In Proceedings of the International AAAI Conference on Web and Social Media, Dublin, Ireland, 4–7 June 2012; Volume 6.
12. Cheng, J.; Adamic, L.; Dow, P.A.; Kleinberg, J.M.; Leskovec, J. Can cascades be predicted? In Proceedings of the 23rd International Conference on World Wide Web, Seoul, Republic of Korea, 7–11 April 2014; pp. 925–936.
13. Lee, J.G.; Moon, S.; Salamatian, K. Modeling and predicting the popularity of online content with Cox proportional hazard regression model. *Neurocomputing* **2012**, *76*, 134–145. [CrossRef]
14. Shen, H.; Wang, D.; Song, C.; Barabási, A.L. Modeling and predicting popularity dynamics via reinforced poisson processes. In Proceedings of the AAAI Conference on Artificial Intelligence, Quebec City, QC, Canada, 27–31 July 2014; Volume 28.
15. Zadeh, A.H.; Sharda, R. Modeling brand post popularity dynamics in online social networks. *Decis. Support Syst.* **2014**, *65*, 59–68. [CrossRef]
16. Li, C.; Ma, J.; Guo, X.; Mei, Q. Deepcas: An end-to-end predictor of information cascades. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 577–586.
17. Cao, Q.; Shen, H.; Cen, K.; Ouyang, W.; Cheng, X. Deephawkes: Bridging the gap between prediction and understanding of information cascades. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 1149–1158.
18. Chen, X.; Zhang, K.; Zhou, F.; Trajcevski, G.; Zhong, T.; Zhang, F. Information diffusion prediction via recurrent cascades convolution. In Proceedings of the 2019 IEEE 35th International Conference on Data Engineering, Macao, China, 8–11 April 2019; pp. 770–781.
19. Akpinar, E.; Berger, J. Valuable virality. *J. Mark. Res.* **2017**, *54*, 318–330. [CrossRef]
20. Tellis, G.J.; MacInnis, D.J.; Tirunillai, S.; Zhang, Y. What drives virality (sharing) of online digital content? The critical role of information, emotion, and brand prominence. *J. Mark.* **2019**, *83*, 1–20. [CrossRef]
21. Dubois, D.; Bonezzi, A.; De Angelis, M. Sharing with friends versus strangers: How interpersonal closeness influences word-of-mouth valence. *J. Mark. Res.* **2016**, *53*, 712–727. [CrossRef]
22. Gao, S.; Ma, J.; Chen, Z. Effective and effortless features for popularity prediction in microblogging network. In Proceedings of the 23rd International Conference on World Wide Web, Seoul, Republic of Korea, 7–11 April 2014; pp. 269–270.
23. Zhang, B.; Qian, Z.; Lu, S. Structure pattern analysis and cascade prediction in social networks. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Riva del Garda, Italy, 19–23 September 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 524–539.
24. Bao, P.; Shen, H.W.; Huang, J.; Cheng, X.Q. Popularity prediction in microblogging network: A case study on sina weibo. In Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013; pp. 177–178.
25. Szabo, G.; Huberman, B.A. Predicting the popularity of online content. *Commun. ACM* **2010**, *53*, 80–88. [CrossRef]
26. Zaman, T.; Fox, E.B.; Bradlow, E.T. A bayesian approach for predicting the popularity of tweets. *Ann. Appl. Stat.* **2014**, *8*, 1583–1611. [CrossRef]
27. Zhao, Q.; Erdogdu, M.A.; He, H.Y.; Rajaraman, A.; Leskovec, J. 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, Sydney, Australia, 10–13 August 2015; pp. 1513–1522.
28. Gao, J.; Shen, H.; Liu, S.; Cheng, X. Modeling and predicting retweeting dynamics via a mixture process. In Proceedings of the 25th International Conference Companion on World Wide Web, Montreal, QC, Canada, 11–15 April 2016; pp. 33–34.
29. Liao, Z.; Lan, P.; Fan, X.; Kelly, B.; Innes, A.; Liao, Z. SIRVD-DL: A COVID-19 deep learning prediction model based on time-dependent SIRVD. *Comput. Biol. Med.* **2021**, *138*, 104868. [CrossRef]
30. Yang, Z.; Guo, J.; Cai, K.; Tang, J.; Li, J.; Zhang, L.; Su, Z. Understanding retweeting behaviors in social networks. In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, Toronto, ON, Canada, 26–30 October 2010; pp. 1633–1636.
31. Kupavskii, A.; Ostroumova, L.; Umnov, A.; Usachev, S.; Serdyukov, P.; Gusev, G.; Kustarev, A. Prediction of retweet cascade size over time. In Proceedings of the 21st ACM International Conference on Information and Knowledge Management, Maui, HI, USA, 29 October–2 November 2012; pp. 2335–2338.
32. Zhao, Y.; Wang, C.; Chi, C.H.; Lam, K.Y.; Wang, S. A Comparative Study of Transactional and Semantic Approaches for Predicting Cascades on Twitter. In Proceedings of the IJCAI, Stockholm, Sweden, 3–19 July 2018; pp. 1212–1218.
33. Raafat, R.M.; Chater, N.; Frith, C. Herding in humans. *Trends Cogn. Sci.* **2009**, *13*, 420–428. [CrossRef]
34. Bikhchandani, S.; Hirshleifer, D.; Welch, I. A theory of fads, fashion, custom, and cultural change as informational cascades. *J. Political Econ.* **1992**, *100*, 992–1026. [CrossRef]
35. Borghol, Y.; Ardon, S.; Carlsson, N.; Eager, D.; Mahanti, A. The untold story of the clones: Content-agnostic factors that impact YouTube video popularity. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, China, 12–16 August 2012; pp. 1186–1194.
36. Gilbert, E. Widespread underprovision on reddit. In Proceedings of the 2013 Conference on Computer Supported Cooperative Work, San Antonio, TX, USA, 23—27 February 2013; pp. 803–808.

37. Lakkaraju, H.; McAuley, J.; Leskovec, J. What's in a name? understanding the interplay between titles, content, and communities in social media. In Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media, Cambridge, MA, USA, 8–11 July 2013.
38. Khosla, A.; Das Sarma, A.; Hamid, R. What makes an image popular? In Proceedings of the 23rd International Conference on World Wide Web, Seoul, Republic of Korea, 7–11 April 2014; pp. 867–876.
39. Chen, J.; Song, X.; Nie, L.; Wang, X.; Zhang, H.; Chua, T.S. Micro tells macro: Predicting the popularity of micro-videos via a transductive model. In Proceedings of the 24th ACM International Conference on Multimedia, Chicago, IL, USA, 22–24 June 2016; pp. 898–907.
40. Ahmed, M.; Spagna, S.; Huici, F.; Niccolini, S. A peek into the future: Predicting the evolution of popularity in user generated content. In Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, Rome, Italy, 4–8 February 2013; pp. 607–616.
41. Abisheva, A.; Garimella, V.R.K.; Garcia, D.; Weber, I. Who watches (and shares) what on youtube? and when? using twitter to understand youtube viewership. In Proceedings of the 7th ACM International Conference on Web Search and Data Mining, New York, NY, USA, 24–28 February 2014; pp. 593–602.
42. Chen, X.; Zhang, K.; Zhou, F.; Trajcevski, G.; Zhong, T.; Zhang, F. Information cascades modeling via deep multi-task learning. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 885–888.
43. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
44. Perozzi, B.; Al-Rfou, R.; Skiena, S. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 701–710.
45. Grover, A.; Leskovec, J. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 855–864.
46. Ribeiro, L.F.; Saverese, P.H.; Figueiredo, D.R. struc2vec: Learning node representations from structural identity. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 385–394.
47. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The graph neural network model. *IEEE Trans. Neural Netw.* **2008**, *20*, 61–80. [CrossRef] [PubMed]
48. Kampffmeyer, M.; Chen, Y.; Liang, X.; Wang, H.; Zhang, Y.; Xing, E.P. Rethinking knowledge graph propagation for zero-shot learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 11487–11496.
49. Beck, D.; Haffari, G.; Cohn, T. Graph-to-sequence learning using gated graph neural networks. *arXiv* **2018**, arXiv:1806.09835.
50. Li, Y.; Tarlow, D.; Brockschmidt, M.; Zemel, R. Gated graph sequence neural networks. *arXiv* **2015**, arXiv:1511.05493.
51. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
52. Hamilton, W.L.; Ying, R.; Leskovec, J. Inductive representation learning on large graphs. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 1025–1035.
53. Jin, X.; Wang, Y. Research on social network structure and public opinions dissemination of micro-blog based on complex network analysis. *J. Netw.* **2013**, *8*, 1543. [CrossRef]
54. Cao, Q.; Shen, H.; Gao, J.; Wei, B.; Cheng, X. Popularity prediction on social platforms with coupled graph neural networks. In Proceedings of the 13th International Conference on Web Search and Data Mining, Houston, TX, USA, 3–7 February 2020; pp. 70–78.
55. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J Mach Learn Res.* **2008**, *9*, 2579–2606.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.