Rumor detection driven by graph attention capsule network on dynamic propagation structures

Peng Yang1,2 · Juncheng Leng1,2 · Guangzhen Zhao1,2 · Wenjun Li1,2 · Haisheng Fang1,2

Accepted: 11 September 2022 / Published online: 12 October 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Rumor detection aims to judge the authenticity of posts on social media (such as Weibo and Twitter), which can effectively prevent the spread of rumors. While many recent rumor detection methods based on graph neural networks can be conducive to extracting the global features of rumors, each node of the rumor propagation structure learned from graph neural networks is considered to have multiple individual scalar features, which are insufficient for reflecting the deep-level rumor properties. To address the above challenge, we propose a novel model named graph attention capsule network on dynamic propagation structures (GACN) for rumor detection. Specifically, GACN consists of two components: a graph attention network enforced by capsule network that can encode static graphs into substructure classification capsules for mining the deep-level properties of rumor, and a dynamic network framework that can divide the rumor structure into multiple static graphs in chronological order for capturing the dynamic interactive features in the evolving process of the rumor propagation structure. Moreover, we use the capsule attention mechanism to combine the capsules generated from each substructure to focus more on informative substructures in rumor propagation. Extensive validation on two real-world datasets demonstrates the superiority of GACN over baselines.

Keywords Rumor detection · Graph neural network · Capsule network · Attention mechanism

✉ Peng Yang
pengyang@seu.edu.cn

Extended author information available on the last page of the article
1 Introduction

The rapid development of social media has changed the way people communicate with each other in daily life and has contributed to the proliferation of many rumors. Rumors quickly and widely spread, and their proliferation pollutes the social network ecology and affects users’ access to high-quality information. Selecting the COVID-19 epidemic in 2020 as an example, the rapid spread of much false information on social networks has caused public panic. Therefore, correctly identifying rumors has become an important research task for scholars and even the industry.

A rumor is defined as a story or statement in general circulation without confirmation or certainty of fact [1]. Consider Fig. 1 as an example, which shows a rumor event on Twitter with a source post and related comments. Previous methods mainly focus on the use of machine learning to detect rumors, such as Decision Tree [2], Random Forest [3], and Support Vector Machine (SVM) [4]. These methods are trained using extracted features that can effectively represent rumors, such as user features, text content, and propagation patterns [2, 4–7]. Such methods heavily rely on feature engineering, which is time-consuming and requires numerous human resources. Moreover, the handcrafted features are highly subjective and cannot capture deep-level features. In recent years, to extract higher-order features, many deep learning techniques have been widely applied in the field of rumor detection. Based on deep learning models, such as CNNs [8–11] and RNNs [12–14], researchers have proposed many models of rumor detection. However, these models fail to take into account the characteristics of the rumor propagation structure. Recent graph models such as GCN [15, 16], GAT [17], and GraphSage [18, 19] have successively emerged, attracting the attention of numerous researchers. Tian et al. [20] proposed a bidirectional graph convolutional network structure, in which the upwards and downwards propagation modes of social media texts were combined to effectively capture the global features of the rumor structure.

Although graph neural networks have been widely employed in rumor detection [20–27], certain problems still need to be solved. When the rumor propagation structure is learned from the graph neural network to the graph embedding, the learning representation of each text node is considered to comprise multiple
individual scalar features rather than an interdependent feature vector, which is not enough to express the deep-level properties. On the other hand, the current methods of rumor detection are mainly applied to a single graph structure and rumors are greatly affected by time, which cannot effectively capture the dynamic interaction information in the propagation process.

To address the above challenges, inspired by capsule network [28, 29], we propose a rumor detection model named graph attention capsule network on dynamic propagation structures (GACN). The model can effectively capture the deep-level properties of each graph node, and combined with the dynamic network framework, it can mine the dynamic interactive features among users in rumor propagation.

Our proposed method consists of two components: the graph attention capsule network and the dynamic network framework. In the graph attention capsule network, first, the graph attention network is applied to the graph-like propagation structure of rumors for aggregation calculation to obtain the globalized features of each graph node. Second, the capsule network is used to convert the scalar values on the graph node features into capsules using its unique dynamic routing mechanism [28] to deeply mine the properties of the texts. The model also incorporates the source post feature to enhance the feature representation of each graph node.

The main task of the dynamic network framework is to divide the comments into a series of substructures in chronological order. We encode each of them by using the graph attention capsule network to capture the interaction features during propagation. Last, the capsule attention mechanism is designed to combine each structure to focus on more informative substructures in the propagation process.

The main contributions of this paper are summarized as follows:

- We propose a new model driven by the graph attention capsule network, GACN (Graph Attention Capsule Network on Dynamic Propagation Structures), to effectively mine the deep-level properties of rumors. To the best of our knowledge, this is the first application of capsule network for rumor detection.
- To capture the dynamic interactive features in the evolving process of the rumor propagation structure, we elaborately design a strategy that divides the dynamic propagation structure of rumor into multiple static graphs in chronological order.
- The GACN model is evaluated on two social network datasets, and the experimental results demonstrate that the proposed method has higher rumor detection performance than other advanced baselines.

The structure of the paper is described as follows. Section 2 introduces relevant and recently related work. Section 3 introduces the problem definition of rumor detection. Section 4 provides the experimental details of our proposed model. Section 5 describes the experimental settings. Section 6 reports and discusses the experimental results. We summarize the findings of the paper in Sect. 7.
2 Related work

The research on rumor detection is mainly divided into two aspects: (1) text content and user information, and (2) propagation structure. The comparison of previous works is summarized in Table 1.

### 2.1 Text content and user information

Previous studies on rumor detection extracted handcrafted features based on the text contents and then applied machine learning methods to classify rumors. Chua et al. [30] used a logistic regression classifier to identify rumors by analyzing six types of features such as comprehensibility, emotion, writing style, and theme of text content. Castillo et al. [2] investigated the information credibility based on handcrafted text features and utilized a decision tree model for the classification task of rumors. These methods are based on machine learning methods that require manual design and extraction of features, which is time-consuming and labor intensive. In recent years, the development of deep learning has provided many new methods for rumor detection. Ma et al. [12] exploited the characteristics of rumor temporal features to divide comments into subblocks of equal length in chronological order, which were input to an RNN for rumor detection. Liu et al. [31] divided source microblogs and related microblogs by cluster analysis and used convolutional neural network (CNN) to identify rumors.

Relying on textual information alone is not effective for rumor detection. In recent years, many studies have considered incorporating user information to assist text content for rumor detection. User-based methods are mainly employed for modeling user participation in social media, where the user’s characteristic information, such as description, gender, followers, friends, location, and

| Papers  | Text | User | Propagation structure |
|---------|------|------|-----------------------|
| [30]    | ✓    | ×    | ×                     |
| [2]     | ✓    | ✓    | ×                     |
| [12]    | ✓    | ×    | ×                     |
| [31]    | ✓    | ×    | ×                     |
| [4]     | ×    | ✓    | ×                     |
| [32]    | ×    | ✓    | ×                     |
| [33]    | ✓    | ×    | ✓                     |
| [34]    | ✓    | ×    | ✓                     |
| [20]    | ✓    | ×    | ✓                     |
| [35]    | ✓    | ×    | ✓                     |

✓ and × mean that the method has or does not have this capability, respectively.
verification type, is collected from the user’s profile. Yang et al. [4] extracted user characteristics for classification, such as gender, geographic location, and the number of followers. Castillo et al. [2] utilized user characteristics on Twitter to detect fake news, which includes the number of followers, number of friends, age of registration, etc. Liu et al. [32] combined RNN and CNN to capture users’ characteristics based on time series to improve the performance of rumor detection.

2.2 Propagation structure

Rumor detection methods based on propagation structure usually analyze the propagation paths or networks formed by retweets and comments of blog posts to identify rumors. Ma et al. [33] proposed a series of time series features based on the life cycle of rumors and applied these features for classification to improve the detection of rumors. Ma et al. [34] used recurrent neural networks to model the top-down propagation direction and the bottom-up diffusion direction of the propagation tree. However, these methods usually focus only on learning serialization features from a propagation perspective, disregarding the global forwarding relationships between posts on social networks. Certain recent studies modeled the propagation of information as a propagation graph and utilized graph neural networks to solve the problems of rumor detection. Bian et al. [20] exploited the graph convolutional network to mine the propagation structure characteristics of the propagation tree based on top-down and bottom-up directions. Yang et al. [35] proposed a rumor detection model based on a graph adversarial learning framework, in which an attacker dynamically adds intentional perturbations to the graph structure to deceive the detector and in which the detector will learn more unique structural features to resist these perturbations, thus improving the robustness and generalization of the model.

Although graph neural networks have achieved good performance for rumor detection [20–27], the learned representation of each text node is considered to comprise multiple individual scalar features rather than a feature vector with interdependencies, which is not sufficient to effectively represent the deep-level properties of rumors. Additionally, existing methods have difficulty capturing the dynamic interaction information during rumor propagation. Therefore, we propose GACN to better address these problems.

3 Problem Definition

Rumor detection is considered an event-level classification task. Let \( c = \{x_1, x_2, ..., x_n, S\} \) be an event, where \( x_j \) is the source post, \( x_j \) represents the \( j-1 \)-th comment, and \( n-1 \) comments are listed in chronological order. As time evolves, the propagation structure of rumors will constantly change, thus forming the propagation structure \( S = \{S^{(1)}, S^{(2)}, ..., S^{(T)}\} \) at different times, where
$T$ is the number of divisions and $S^{(i)}$ is the $i^{th}$ propagation substructure. The details of the delineation are described in the Method.

Each event $c \in C$ can be labeled with a category label $y \in Y$. The task of rumor detection is to learn the function $f$ from the event set $C$ to the label set $Y$. It is formulated as:

$$f : C \rightarrow Y$$

(1)

4 Method

In this section, we introduce the overall framework of GACN and its detail of implementation. Figure 2 shows an overview of our framework, which will be described in the following sections. Table 2 summarizes some important notations used in this paper. We propose a Graph Attention Capsule Network on Dynamic Propagation Structures, named GACN. This model consists of two components: the graph attention capsule network which mines the deep-level properties of rumors and the dynamic network framework which captures the dynamic interactive features in the evolution process of the rumor propagation structure.

Fig. 2 Illustration of our GACN model. It consists of two components: the dynamic network framework (left) and the graph attention capsule network (right)
4.1 Graph Attention Capsule Network

We design the graph attention capsule network to obtain the deep-level properties of each propagation structure inspired by capsule network [28]. The network consists of three parts: global features of nodes, source post encoding, and substructure classification capsules. We first describe how to generate the global features of nodes.

4.1.1 Global features of nodes

We construct graph propagation structure $< V, E >$ for each rumor $c$, depending on the response relationships between comments and between comments and source post. $V^{(t)} = \{ x_1, x_2, ..., x_{d^{(t)}} \}$ represents the nodes in the $t - th$ propagation substructure $S^{(t)} = < V^{(t)}, E^{(t)} >$, including $x_1$ being the node of the source post, $x_i$ being the node of the comments, and $E^{(t)}$ representing the set of edges that describe the response relationship between nodes. For each textual content, we encode them using the TF-IDF model [36].
First, the affine transformation matrix $\mathbf{B}$ transforms the initial feature $x_i$ into a hidden vector $h_i^{(0)}$, as shown in Eq. 2. The GAT, proposed by Veličković et al. [17], is then applied to the hidden vectors to obtain the global features of each node. In updating the feature vectors of graph nodes, the graph attention network aggregates the neighbor information according to the weights to obtain the representation that contains the neighbor information. The detailed steps are presented as follows:

$$h_i^{(0)} = \mathbf{B}x_i \quad (2)$$

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(a^T[\mathbf{W}h_i^{(0)}\|\mathbf{W}h_j^{(0)})])}{\sum_{k \in N_i} \exp(\text{LeakyReLU}(a^T[\mathbf{W}h_i^{(0)}\|\mathbf{W}h_k^{(0)}]))} \quad (3)$$

$$h_i^{(l+1)} = \sigma(\sum_{j \in N_i} \alpha_{ij} \mathbf{W}h_j^{(l)}) \quad (4)$$

The $\alpha_{ij}$ obtained by Eq. 3 represents the weight of importance when the features of the node $j$ are aggregated to node $i$, where $N_i$ is the first-order neighbors of node $i$, both $\mathbf{W}$ and $a$ are trainable parameters, $h_i^{(0)}$ is the representation of the $i$-th graph node after $l$ graph attention layers are sequentially computed, and $\text{LeakyReLU}$ is the activation function. By summing the vectors of neighbors according to the weights $\alpha_{ij}$, we can obtain the vector $h_i^{(l+1)}$ of node $i$ at layer $l+1$ of the graph, as shown in Eq. 4. We concatenate the vectors $\mathbf{H}_i$ obtained from the output of each graph attention layer for node $i$ to obtain the global vector $\mathbf{H} \in \mathbb{R}^{N \times p \times d_m}$ on the substructure $S^{(t)}$. It is formulated as:

$$\mathbf{H}_i = \text{concat}([h_i^{(1)}, h_i^{(2)}, \ldots, h_i^{(p)})]) \quad (5)$$

$$\mathbf{H} = \text{concat}([\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_N]) \quad (6)$$

where $p$ is the number of graph network layers, $N$ denotes the number of graph nodes, and $d_m$ is the dimension of the hidden vector after each graph attention layer is encoded.

### 4.1.2 Source post encoding

The source post contains rich information about a rumor, which is beneficial to strengthen the feature representation of graph nodes. In our proposed model, the transformer encoder [37] is used to encode the source post.

First, we use GloVe [38] to generate a word vector $\{w^1_r, w^2_r, \ldots, w^{n'}_r\}$ for each word in the source post $x_1$, where $n'$ refers to the number of words. Second, the self-attention mechanism is used to measure the importance of words. It is formulated as:

$$h_r = \text{TransformerEncoder}(\{w^1_r, w^2_r, \ldots, w^{n'}_r\}) \quad (7)$$
where $h_r$ is the result of the encoding by the encoder module in the transformer. Last, the final source post feature $r \in \mathbb{R}^{d_r}$ is obtained by averaging the hidden vectors of all words, where $d_r$ is the dimension of the source post feature.

### 4.1.3 Substructure classification capsules

After the aggregation calculation of each node through the graph attention network, the obtained vector $H$ indicates that it has been able to capture the global features of the propagation structure. Considering that the source post of the rumor can highlight the typical features of the rumor, the model concatenates the source post feature with the global features of each node to obtain the representation $A \in \mathbb{R}^{N \times p \times (d_m + d_r)}$ of each graph node which is enhanced. We use the one-dimensional convolution $conv1d(\bullet)$ by concatenating the features of different graph network layers at the same location to obtain the primary capsules $H' \in \mathbb{R}^{N \times q \times d_c}$, where $q$ is the number of primary capsules and $d_c$ is the dimension of primary capsules. It is formulated as:

$$A = concat(H, r)$$ (9)  

$$H' = conv1d(A)$$ (10)

Substructure classification capsules are obtained by transforming primary capsules, which indicate the type of rumors. Each substructure classification capsule is a vector, and the scalar values in the vector represent the deep-level properties of the rumor.

Node normalization is performed to generate the attention value $a_{primary}$ which is applied to the primary capsules. To obtain the normalized primary capsules $U \in \mathbb{R}^{N \times q \times d_c}$, we employ two fully connected layers. It is formulated as:

$$a = FC_2(FC_1(H'))$$ (11)  

$$U = a_{primary} \ast H'$$ (12)

The dynamic routing mechanism in the capsule network [28] is employed to transform primary capsules into substructure classification capsules. In the algorithm, $b_{ij}$ is initialized and the coupling coefficient weight $m_i$ is obtained, which represents the contribution of the normalized primary capsules $i$ to the substructure classification capsules. It is formulated as:

$$b_{ij} = b_{ij} + u_{j|i} \ast v_j$$ (13)  

$$m_i = softmax(b_i)$$ (14)

where $b_{ij}$ and $u_{j|i}$ denote the training parameters and prediction vectors, respectively, of the normalized primary capsule $i$ to the substructure classification capsule $j$. The
prediction vector $u_{ji}$ is obtained by the weight $w_{ij}$ of the normalized primary capsules $U$. The initialized capsule weight $m_{ij}$ is used to obtain the intermediate value $s_j$ of the substructure classification capsule $j$. $s_j$ is calculated by the activation function Squashing [28] to obtain the output $v_j$ of the substructure classification capsule $j$. The weight $m_{ij}$ is iteratively updated through the prediction vector $u_{ji}$ and capsule output $v_j$. It is formulated as:

$$u_{ji} = w_{ij}U \quad (15)$$

$$s_j = \sum m_{ij} \ast u_{ji} \quad (16)$$

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|^2} \quad (17)$$

**4.2 Dynamic Network Framework**

**4.2.1 Partition propagation structure**

To capture the dynamic interaction characteristics in the evolving process of rumor propagation structure, we divide the structure according to the release time of each post. Exactly, all comments below each source post are divided by equal amounts. The propagation substructure starts with the number $S^{(1)}$ and then separately adds $\lceil(n - 1)/T \rceil$ comments to form the next substructure, where $n - 1$ is the number of comments and $T$ is the number of partitions. The last substructure is formed when the number of comments increases to $T \times \lceil(n - 1)/T \rceil$. The propagation structure $S$ representing the event $c$ is described as:

$$S = \{S^{(1)}, S^{(2)}, \ldots, S^{(T)}\} \quad (18)$$

**4.2.2 Capsule attention mechanism**

To effectively capture the dynamic propagation characteristics of rumors, after the capsule network obtains the classification capsule vectors $G^{(i)} = [v_1, v_2, \ldots, v_k]$ of the substructure $S^{(i)}$, where $k$ is the number of rumor classes, we use the self-attention mechanism to measure the importance of the classification capsules of each substructure as shown in Fig. 3.

The capsules belonging to the same category in all structures are formed into a matrix as the common values of $Q$, $K$, and $V$, where $Q$, $K$, and $V$ are the query vector, key vector and value vector, respectively. Therefore, the attentional substructure classification capsules $h_{G^{(i)}}$ are calculated according to Eq. 19, where $d_K$ is used to keep the gradient stable. The final classification capsules $I \in \mathbb{R}^{k \times f}$ are obtained by averaging

$$Springer$$
over all substructure vectors $h_{G^0}$, where $k$ is the number of categories and $f$ is the dimension of the final classification capsules. It is formulated as:

$$Attention(Q,K,V) = \text{softmax}(\frac{(QK^T)}{\sqrt{d_k}})V$$ (19)

$$I = \frac{1}{T} \sum_{i=1}^{T} h_{G^0}$$ (20)

where $T$ is the number of partitions and $\text{softmax}(\bullet)$ is the activation function.

4.3 Prediction and classification

In this paper, the margin loss function [28] is used as the loss function of our proposed model. It can be expressed as Eq. 21:

$$\text{Loss}_c = \sum_{i} \left\{ \hat{T}_i \max (0, m^+ - ||I_i||)^2 + \lambda (1 - \hat{T}_i) \min (0, ||I_i|| - m^-)^2 \right\}$$ (21)

where $i$ is the type of rumor, $||I_i||$ is the output probability of final classification capsule $i$, $\hat{T}_i$ is the indicator function of classification (the existence of class $i$ is 1, and the nonexistence is 0), and $m^+$ is the upper bound, penalizing false positives, i.e., predicting the presence of class $i$ but its true absence. $m^-$ is the lower bound, penalizing false negatives, i.e., predicting the absence of class $i$ but its true presence, and $\lambda$ is the scaling factor, adjusting the weight of both. The overall algorithm description of our GACN is shown in Algorithm 1.
Experiment

In this section, we first introduce the two mainstream datasets used in the experiment, including Twitter15 [39] and Twitter16 [33], and then describe the baseline models selected in this article that contrast with the proposed method. Next, the experimental evaluation indicators are described. Finally, we describe the parameter settings of our proposed model.
5.1 Dataset

This paper will evaluate our proposed method on two publicly available datasets, including Twitter15 [39] and Twitter16 [33]. The statistics of the datasets are shown in Table 3.

Twitter15 and Twitter16 were created by Ma et al. [33, 39], who collected rumor information from the Twitter website, a famous international social networking platform at different times. The authors collect 1490 and 818 rumors that are marked into four categories: true rumor (TR), false rumor (FR), unverified rumor (UR), and nonrumor (NR). The method of dividing the Twitter15 and Twitter16 datasets refers to existing research [34], and the experiment is carried out using a fivefold cross-validation method.

5.2 Baselines

In this paper, certain state-of-the-art methods for rumor detection are selected as the baseline models to compare with our proposed model. These models are presented as follows:

- DTC [2]: A rumor detection model based on global handcrafted features to build a decision tree classifier to obtain information credibility.
- SVM-RBF [4]: A support vector machine classifier based on RBF kernel which uses statistical features manually constructed from blog content.
- SVM-TS [33]: A linear support vector machine classifier based on temporal context features.
- SVM-TK [39]: An SVM classifier with a propagation tree kernel on the basis of the propagation structures of rumors.
- GRU-RNN [12]: An RNN-based model that captures contextual information from continuous representations of relevant posts over time.

| Statistic           | Twitter15 | Twitter16 |
|---------------------|-----------|-----------|
| # of posts          | 331612    | 204820    |
| # of True rumors    | 374       | 205       |
| # of False rumors   | 370       | 205       |
| # of Unverified rumors | 374   | 203       |
| # of Nonrumors      | 372       | 205       |
| # of events         | 1490      | 818       |
| Avg. time length/event | 1,337 Hours | 848 Hours |
| Avg. # of posts/event | 223     | 251       |
| Max # of posts/event | 1768     | 2765      |
| Min # of posts/event | 55       | 81        |
- RvNN [34]: A model based on RNN models to model the propagation direction and the diffusion direction, respectively, to learn the feature vector representation of the propagation tree.
- GLAN [21]: A model that combines global and local features to build heterogeneous graphs and extract features.
- GCAN [23]: A GCN-based model that can describe the rumor propagation mode and use the dual co-attention mechanism to capture the relationship among source post, user characteristics and propagation path.
- Bi-GCN [20]: A graph convolution neural network model based on the propagation direction and diffusion direction of the propagation tree.
- P-BiGAT [22]: A bidirectional graph attention networks based on the propagation tree and diffusion tree through the tweet comment and reposting relationship.

5.3 Evaluation metrics

Rumor detection investigated in this paper is essentially a classification problem, so we choose classification-based evaluation metrics to evaluate the performance. The accuracy (Acc) and the F1-value (F1) of each category are chosen as the evaluation metrics in the paper, and the calculation formulas are as:

\[
Acc = \frac{TP + FN}{TP + FP + TN + FN}
\]  
(22)

\[
F1 = \frac{2 \times TP}{2 \times TP + FP + FN}
\]  
(23)

where TP (true positive) denotes the number of cases in which the true category is positive, and the predicted category is positive, FP (false positive) denotes the number of cases in which the true category is negative and the predicted category is positive, FN (false negative) denotes the number of cases in which the true category is positive and the predicted category is negative, and TN (true negative) denotes the number of cases in which the true category is negative and the predicted category is negative.

5.4 Experimental settings

Our experiments are implemented with PyTorch. The optimizer is Adam [40] with a learning rate of 0.001. Our model uses TF-IDF to initialize the text nodes, and its dimension is 5000. We adopt Glove with a dimension of 200 to encode the source post. The model is trained by using 3 layers of graph attention network layers and the hidden vector dimension between each layer is 64. The rate of dropout is 0.5. The experiments are set up with an early stopping strategy, which is to terminate the training when the accuracy of the validation set no longer decreases within 10 iterations. A 5-fold cross-validation method is applied to evaluate the experimental
effects, and the results are averaged as the metrics of the model. The hyperparameters involved in the experiment are listed in Table 4.

### 6 Experimental results

In this section, we validate our proposed model in three aspects: rumor detection results, ablation study, and visualization.

#### 6.1 Rumor detection results

Table 5 and Table 6 record the detection effects of the state-of-the-art models on Twitter15 and Twitter16. Among them, we rerun the open-source code provided by
Bian et al. [20] and Lu et al. [23] to obtain the experimental results of Bi-GCN and GACN.

From the results in Table 5 and Table 6, we can determine that our proposed GACN model outperforms the state-of-the-art baselines on both Twitter15 and Twitter16. GACN improves 1.9% in accuracy compared with the optimal baseline model on Twitter15, while improving 0.4% on Twitter16. The results indicate that our proposed model has much better detection performance.

First, as shown in Table 5 and Table 6, all models based on deep learning (GRU-RNN, RvNN, GLAN, GCAN, Bi-GCN, P-BiGAT, and GACN) are superior to most of traditional models based on handcrafted features. This result confirms that deep learning methods are more accurate than traditional methods in extracting the deeper features of rumors.

Second, compared with RvNN, GCAN, Bi-GCN, P-BiGAT, and GACN are slightly better in all metrics. The main reason is that RvNN is a tree-based model that cannot capture long-distance dependencies in sequences. The graph-based models can solve this problem by capturing global characteristics in rumors, which improves the detection of rumors.

Last, in the graph-based models, the accuracy of GACN is higher than that of other graph-based models, which is attributed to the notion that these models only handle a single static graph, and cannot capture the dynamic interaction characteristics of rumors in the propagation process. On the other hand, these models can only extract the scalar features of graph nodes, which is not enough to effectively mine the deep-level properties of rumors.

### Table 6  Rumor detection results on Twitter16 (%)

| Model   | Acc | F1  |
|---------|-----|-----|
|         | NR  | FR  | TR  | UR  |
| DTC     | 46.5| 64.3| 39.3| 41.9| 40.3|
| SVM-RBF | 55.3| 67.0| 8.5 | 11.7| 36.1|
| SVM-TS  | 57.4| 75.5| 42.0| 57.1| 52.6|
| SVM-TK  | 66.2| 64.3| 62.3| 78.3| 65.5|
| GRU-RNN | 63.6| 61.7| 71.5| 57.7| 52.7|
| RvNN    | 73.7| 66.2| 74.3| 83.5| 70.8|
| GCAN    | 85.9| 76.1| 85.0| 92.6| 85.5|
| GLAN    | 87.4| 85.1| 87.7| 91.3| 85.8|
| Bi-GCN  | 86.8| 78.0| 90.1| 93.4| 85.7|
| P-BiGAT | 89.6| 81.5| **93.4**| **97.9**| 83.4|
| GACN    | **90.0**| **96.4**| 88.9| 90.0| 84.6|

The values in bold indicate the maximum value under different indicators.
6.2 Ablation study

To highlight the experimental effects of our proposed model, we conduct ablation analysis with the following models. The experimental results are shown in Table 7.

- **w/o Dynamic**: This is a static graph capsule network model, not including dynamic network framework.
- **w/o Root**: This model only considers the rumor propagation structure and does not consider the source post.
- **w/o Capsule**: This model replaces the capsule network after the graph network model with a fully connected layer and applies cross-entropy as the loss function.
- **w/o Capsule & w/o Dynamic**: The model does not take into account the capsule network and dynamic network framework.

The results in Table 7 show that our proposed model has the highest accuracy on Twitter15 and Twitter16, while the w/o Capsule & w/o Dynamic achieves poor detection performance. The accuracy of GACN, w/o Root, and w/o Dynamic outperform the w/o Capsule and w/o Capsule & w/o Dynamic without the capsule network, respectively, which confirms that the capsule network can better excavate the deep-level properties of rumors. Second, compared with w/o Root, our proposed model achieves better results, indicating that the information of the source post feature has an important role in improving the performance of rumor detection. Table 7 shows that the GACN and w/o Capsule models outperform the corresponding w/o Dynamic and w/o Capsule & w/o Dynamic without a dynamic network framework in terms of accuracy metrics, which further demonstrates the effectiveness of the dynamic propagation structure. The results suggest the necessity of capturing the dynamic interactive features in the evolving process of the rumor propagation structure. In summary, the dynamic propagation structure of rumors and the capsule network have a certain auxiliary effect for rumor detection.

| Model                          | Twitter15 |       | Twitter16 |       |
|-------------------------------|-----------|-------|-----------|-------|
|                               | Acc       | F1    | Acc       | F1    |
|                               | NR | FR   | TR   | UR   | NR | FR   | TR   | UR   |
| w/o Dynamic                   | 88.5       | 82.9  | 87.9  | **95.0** | 88.7 | 82.5  | 91.4  | **96.3** |
| w/o Root                      | 87.5       | **92.3** | 89.0  | 85.3  | 83.5 | 88.8  | 98.8  | 87.8  |
| w/o Capsule                   | 87.2       | 91.6  | 87.7  | 84.7  | 84.7 | 87.6  | 91.4  | **93.5** |
| w/o Capsule & w/o Dynamic    | 85.4       | 80.3  | 86.9  | 92.7  | 81.9 | 87.0  | 78.1  | 88.9  |
| GACN                          | **88.9**   | 91.0  | **92.4** | 87.5  | 84.5 | **90.0** | **96.4** | 88.9  |

The values in bold indicate the maximum value under different indicators.

Table 7  Effect of different components in GACN (%)
6.3 Visualization

GACN mines the deep-level properties of the rumor by using the capsule network. To clarify the practical effect of the properties in each capsule, we use T-SNE [41] to extract the values of each channel in the final classification capsules for dimensionality reduction and then present them in the form of visualization in Table 8. For Table 8, the categories of rumors are labeled with 0, 1, 2, and 3, representing nonrumor, false rumor, true rumor, and unverified rumor, and the corresponding points in the figures are colored red, green, yellow, and blue, respectively. In each row of the table, we compare two categories, three categories, and four categories of rumors. Selecting the first Row 0_1 as an example, we compare nonrumor and false rumors. Considering the size limitation of the paper, we choose channels that can highlight the properties of rumors for analysis.

Table 8  Visualization of channels in classification capsules

| Categories | Channels | Channel0 | Channel1 | Channel2 | Channel3 | Channel5 |
|------------|----------|----------|----------|----------|----------|----------|
| 0_1        |          |          |          |          |          |          |
| 0_2        |          |          |          |          |          |          |
| 1_2        |          |          |          |          |          |          |
| 1_3        |          |          |          |          |          |          |
| 2_3        |          |          |          |          |          |          |
| 0_2_3      |          |          |          |          |          |          |
| 0_1_2_3    |          |          |          |          |          |          |
Table 8 shows that when the two categories are compared, the vectors of each channel can distinguish specific categories after dimensionality reduction. For example, Channel3 can well distinguish 1 from 2, but it is difficult to distinguish 1 from 3. In contrast, Channel1 can easily distinguish 1 and 3, but not 1 and 2. This result indicates that the properties captured by Channel1 may be the factor determining the difference between 1 and 2, as well as Channel3. Channel2 can almost completely distinguish 0 from 2 and 2 from 3, which also explains why the types of rumors represented by 0, 2, and 3 can be completely divided when comparing the three categories of rumors 0_2_3. In the four classification rows, none of the individual channels can effectively classify each category, so it is necessary to integrate all channels to obtain their respective properties for classification. Compared with scalar-based deep learning models, using the capsule network can better integrate the properties captured by each channel to improve the accuracy of rumor detection.

Next, we use Tsne to reduce the dimensionality of each classification capsule, and the obtained results are shown in Table 9. Almost every classification capsule can roughly distinguish different types of rumors, indicating that each classification capsule focuses on the properties of rumors in different categories. Compared with the scalar-based neural network, the graph structure of rumors can be represented as multiple graph embeddings, each of which contains rich properties, so the use of capsule network can fully extract the features of rumors.

### 7 Conclusion

In this paper, we propose a Graph Attention Capsule Network on Dynamic Propagation Structures, named **GACN**. The method divides the propagation structure by using the dynamic network framework to capture the dynamic interaction...
characteristics in the evolving process of rumor propagation. The graph attention capsule network is then applied to transform the divided static graph structures into capsules to extract deep-level properties. To the best of our knowledge, this is the first model that applies capsule network for rumor detection. Through experiments on two mainstream datasets, compared with the state-of-the-art baselines, the extensive results show that GACN achieves improvements of up to 1.9% and 0.4%, respectively. In future work, we will explore the specific meanings on which each channel in the capsule focuses to visually explain the improvement in rumor detection.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grant 62272100 and in part by the Fundamental Research Funds for the Central Universities and the Academy-Locality Cooperation Project of Chinese Academy of Engineering under Grant JS2021ZT05.

Data availability The datasets analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no competing interests.

Consent to participate All authors contributed to this work.

Consent for Publication All authors have checked the manuscript and have agreed to the submission.

Ethics approval All authors read and approved the final version of the manuscript.

References

1. DiFonzo N, Bordia P (2007) Rumor psychology: Social and organizational approaches
2. Castillo C, Mendoza M, Poblete B (2011) Information credibility on twitter. In: Proceedings of the 20th International Conference on World Wide Web,P 675–684
3. Kwon S, Cha M, Jung K, Chen W, Wang Y (2013) Prominent features of rumor propagation in online social media. In: 2013 IEEE 13th International Conference on Data Mining, pp. 1103–1108 . IEEE
4. Yang F, Liu Y, Yu X, Yang M (2012) Automatic detection of rumor on sina weibo. In: Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics, pp. 1–7
5. Qazvinian V, Rosengren E, Radev D, Mei Q (2011) Rumor has it: Identifying misinformation in microblogs. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pp. 1589–1599
6. Wang AH (2010) Don’t follow me: Spam detection in twitter. In: 2010 International Conference on Security and Cryptography (SECRYPT), pp. 1–10 . IEEE
7. Ratkiewicz J, Conover M, Meiss M, Gonçalves B, Patil S, Flammini A, Menczer F (2010) Detecting and tracking the spread of astroturf memes in microblog streams. arXiv preprint arXiv:1011.3768
8. Yu F, Liu Q, Wu S, Wang L, Tan T et al (2017) A Convolutional Approach for Misinformation Identification. In: IJCAI, pp. 3901–3907
9. Yu F, Liu Q, Wu S, Wang L, Tan T (2019) Attention-based convolutional approach for misinformation identification from massive and noisy microblog posts. Comput Secur 83:106–121
Rumor detection driven by graph attention capsule network…

10. Azri A, Favre C, Harbi N, Darmont J, Noûs C (2021) Calling to cnn-lstm for rumor detection: A deep multi-channel model for message veracity classification in microblogs. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 497–513. Springer

11. Peng Y, Wang J (2021) Rumor detection based on attention cnn and time series of context information. Future Internet 13(11):1–18

12. Ma J, Gao W, Mitra P, Kwon S, Jansen BJ, Wong K-F, Cha M (2016) Detecting rumors from microblogs with recurrent neural networks. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, pp. 3818–3824

13. Ajao O, Bhowmik D, Zargari S (2018) Fake news identification on twitter with hybrid cnn and rnn models. In: Proceedings of the 9th International Conference on Social Media and Society, pp. 226–230

14. Asghar MZ, Habib A, Habib A, Khan A, Ali R, Khattak A (2021) Exploring deep neural networks for rumor detection. J Ambient Intell Human Comput 12(4):4315–4333

15. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, pp. 1–14

16. Defferrard M, Bresson X, Vandergheynst P (2016) Convolutional neural networks on graphs with fast localized spectral filtering. Adv Neural Inform process syst 29:3844–3852

17. Velickovic P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y (2018) Graph attention networks. stat 1050:4

18. Hamilton WL, Ying R, Leskovec J (2017) Inductive representation learning on large graphs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 1025–1035

19. Xu K, Hu W, Leskovec J, Jegelka S (2018) How powerful are graph neural networks? arXiv preprint arXiv:1810.00826

20. Bian T, Xiao X, Xu T, Zhao P, Huang W, Rong Y, Huang J (2020) Rumor detection on social media with bi-directional graph convolutional networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 549–556

21. Yuan C, Ma Q, Zhou W, Han J, Hu S (2019) Jointly embedding the local and global relations of heterogeneous graph for rumor detection. In: 2019 IEEE International Conference on Data Mining (ICDM), pp. 796–805. IEEE

22. Yang X, Ma H, Wang M (2022) Rumor detection with bidirectional graph attention networks. Secur Commun Netw 2022:1–13

23. Lu Y-J, Li C-T (2020) Gcan: Graph-aware co-attention networks for explainable fake news detection on social media. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 505–514

24. Song Y-Z, Chen Y-S, Chang Y-T, Weng S-Y, Shuai H-H (2021) Adversary-aware rumor detection. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp. 1371–1382

25. Li J, Bao P, Shen H, Li X (2021) Mistr: A multiview structural-temporal learning framework for rumor detection. IEEE Transact Big Data 01:1–13

26. Li C, Peng H, Li J, Sun L, Lyu L, Wang L, Yu PS, He L (2022) Joint stance and rumor detection in hierarchical heterogeneous graph. IEEE Transact Neural Netw Learn Syst 33(6):2530–2542. https://doi.org/10.1109/TNNLS.2021.3114027

27. Ran H, Jia C, Zhang P, Li X (2022) Mgt-esp: multi-channel graph attention neural network with event-sharing module for rumor detection. Inform Sci 592:402–416

28. Sabour S, Frosst N, Hinton GE (2017) Dynamic routing between capsules. In: Proceedings of the 31st international conference on neural information processing systems, pp. 3859–3869

29. Xinyi Z, Chen L (2018) Capsule graph neural network. In: International Conference on Learning Representations, pp. 1–16

30. Chua AY, Banerjee S (2016) Linguistic predictors of rumor veracity on the internet. In: Proceedings of the International MultiConference of Engineers and Computer Scientists, vol. 1, pp. 387–391

31. Liu Z, Wei Z, Zhang R (2017) Rumor detection based on convolutional neural network. J Comput Appl 37(11):3053–3056

32. Liu Y, Wu Y-FB (2018) Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence, pp. 354–361
33. Ma J, Gao W, Wei Z, Lu Y, Wong K-F (2015) Detect rumors using time series of social context information on microblogging websites. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 1751–1754
34. Ma J, Gao W, Wong K-F (2018) Rumor detection on twitter with tree-structured recursive neural networks. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1980–1989
35. Yang X, Lyu Y, Tian T, Liu Y, Liu Y, Zhang X (2021) Rumor detection on social media with graph structured adversarial learning. In: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, pp. 1417–1423
36. Salton G, Buckley C (1988) Term-weighting approaches in automatic text retrieval. Inform process manage 24(5):513–523
37. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008
38. Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543
39. Ma J, Gao W, Wong K-F (2017) Detect rumors in microblog posts using propagation structure via kernel learning. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 708–717
40. Kingma DP, Ba J (2015) Adam: A method for stochastic optimization. In: ICLR (Poster), pp. 1–15
41. van der Maaten L, Hinton G (2008) Visualizing data using t-sne. J Mach Learn Res 9:2579–2605

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Peng Yang1,2 · Juncheng Leng1,2 · Guangzhen Zhao1,2 · Wenjun Li1,2 · Haisheng Fang1,2

Juncheng Leng
lengjuncheng@seu.edu.cn

Guangzhen Zhao
zhaogz@seu.edu.cn

Wenjun Li
kerwen@seu.edu.cn

Haisheng Fang
haisheng@seu.edu.cn

1 Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing, China
2 School of Computer Science and Engineering, Southeast University, Nanjing, China