Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

MHAGCNN: A Multi-Head Attention Graph Convolutional Network for Public Opinion Prediction in Social Networks

Yongqing Yang

yangyq@sdtbu.edu.cn

Shandong Technology and Business University

Chenghao Fan

Nanjing University of Aeronautics and Astronautics

Linhan Ouyang

Nanjing University of Aeronautics and Astronautics

Research Article

Keywords: Online social networks, Public opinion popularity prediction, Graph convolutional networks, Attention mechanisms, Bayesian optimization

Posted Date: June 23rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9848989/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

MHAGCNN: A Multi-Head Attention Graph Convolutional Network for Public Opinion Prediction in Social Networks

Yongqing Yang[1*], Chenghao Fan[2] and Linhan Ouyang[2,3]

[1*]Shandong Technology and Business University, School of Management Science and Engineering, Yantai, 264000, China.
[2]Nanjing University of Aeronautics and Astronautics, College of Economics and Management, Nanjing, 211106, China.
[3]Humanities and Social Sciences Laboratory of Jiangsu Province for Digital Intelligence Management and Low-carbon Operations for Manufacturing System, Nanjing University of Aeronautics and Astronautics, Nanjing, 211106, China.

*Corresponding author(s). E-mail(s): yangyq@sdtbu.edu.cn;
Contributing authors: fanchenghao@nuaa.edu.cn; ouyang@nuaa.edu.cn;

Abstract

The rapid growth of mobile communication technologies and social media platforms has increased the complexity of online public opinion dynamics by accelerating information diffusion and strengthening closed-loop interactions within social circles. Accelerated dynamics, compounded by structural opacity and cognitive homogenization, hinder accurate prediction of opinion cascades and timely monitoring of large-scale public opinion events. In this study, we construct Multi-Head Attention Graph Convolutional Neural Networks (MHAGCNN), addressing two fundamental challenges: 1) the integration of dynamic propagation patterns with evolving network topologies and 2) the construction of a graph-based predictive framework that jointly uses structural, temporal, and semantic features for public opinion popularity prediction. Evaluated on a large-scale Weibo dataset containing 112,209 comment records, user attributes, and approximately 67,000 interaction edges, MHAGCNN achieves an $\boldsymbol{R^2}$ score of 0.967 in the final baseline comparison and an average $\boldsymbol{R^2}$ score of 0.965 in five-fold cross validation. Key methodological contributions include the integration of GAT, GraphSAGE, and TransformerConv operators for learning node representations from interaction-based social graphs. The integration of Bayesian hyperparameter optimization reduces manual tuning bias and enhances model generalization. Practically, MHAGCNN can support social media analysis tasks such as public opinion popularity prediction, information diffusion monitoring, and influential-user identification. This research intersects computational social science with graph machine learning, contributing to theoretical advancements in complex network analysis while developing deployable tools for digital governance within hyperconnected social ecosystems.

Keywords: Online social networks, Public opinion popularity prediction, Graph convolutional networks, Attention mechanisms, Bayesian optimization

1 Introduction

Recent years, the rapid development of the Internet, including the continuous change of communication technology and the continued promotion of the intelligent process of online social networking, has brought new challenges to the research and judgment of online public opinion. The global proliferation of 5G networks, with penetration rates exceeding 40%, and the exponential growth of social media platforms, now encompassing 2.8 billion active users, have significantly amplified the complexity of online opinion dynamics. This technological advancement has simultaneously reshaped the architecture of social networks, leading to closed-loop social circles characterized by internal interactions and collective unconsciousness [45], which exhibit faster information dissemination speeds than general communities. This accelerated propagation, coupled with structural opacity and cognitive homogenization within these circles, poses critical challenges in predicting opinion cascades and mitigating large-scale public opinion crises. Although recent advances in the temporal modeling and structural analysis frameworks for social networks have improved prediction accuracy, three critical gaps persist. First, existing approaches do not adequately model time-varying topological dynamics in social circles. Second, they oversimplify spatiotemporal dependencies in propagation paths. Third, current methods insufficiently integrate structural, temporal, and semantic features into a unified graph-based predictive framework.

This study addresses two fundamental questions:

1. How can dynamic interaction patterns and graph topology be jointly modeled for public opinion popularity prediction?
2. How can structural, temporal, and semantic features be integrated into a graph neural network to improve predictive robustness?

We propose Multi-Head Attention Graph Convolutional Neural Networks (MHAGCNN), a graph convolutional neural network that integrates different convolutional layers with a multi-head attention structure, featuring three technical breakthroughs:

- Feature-integrated graph construction: structural, temporal, and semantic features are extracted from user attributes, interaction records, and text embeddings.
- Sparse adjacency representation: interaction strength and semantic similarity are combined to construct a weighted graph, which is converted into a sparse edge index for efficient graph learning.
- Hybrid graph neural architecture: GAT, GraphSAGE, and TransformerConv layers are integrated to capture local neighbor importance, neighborhood aggregation, and higher-order contextual dependencies.

Evaluated on a Weibo dataset, MHAGCNN achieves the best performance among the compared models, with MSE, RMSE, and R^2 values of 0.129, 0.359, and 0.967, respectively. Five-fold cross validation further shows stable predictive performance, with an average R^2 of 0.965. These results indicate that the proposed architecture can effectively capture structural, temporal, and semantic dependencies in social media networks.

Section 3 formalizes the feature construction pipeline and the MHAGCNN architecture. Section 4 reports the experimental setting, cross-validation results, Bayesian optimization process, and baseline comparisons. Section 5 concludes the study and discusses theoretical and practical implications.

This work bridges computational social science and graph machine learning, offering theoretical insights for complex network analysis and actionable tools for digital governance.

2 Literature Review

2.1 Evolution of Online Opinion Dynamics Research

Research on dynamic modeling of online public opinion is mainly based on the cross-disciplinary integration of computational social sciences [1] and complex system theory [2]. Early classic models such as the Deffuant model [3, 4], the Sznajd model [5, 6] and the SIR (Susceptible, Infected and Recovered) topic model [7, 8] were grounded in the assumption of bounded rationality, simulating the convergence process of group opinions through individual viewpoint iteration. However,

these models rely on static network structures and linear interaction assumptions, making it difficult to characterize the non-linear propagation induced by closed-loop interaction in social circles. The opinion dynamics framework [9] simulates the evolution of macro public opinion through preset individual interaction rules, but its core assumption is that the network topology remains static and the propagation process follows linear diffusion [10], which may lead to opacity in the structure of social circles, such as insufficient ability to capture hidden connections in small groups, and neglect of the amplification effect of cognitive homogenization on opinion polarization [11].

In capturing the non-linear dynamics and hidden social circle interactions, evaluating public opinion evolution necessitates integrating structural and cognitive attributes of social circles into the framework. In evaluating the scale and level of events of public opinion, researchers considered factors such as opinion trend [12, 13], topic relevance [14, 15], and public opinion influence [16, 17] to establish an objective evaluation index system. However, the structure complexity of social circle drives the development of public opinion [18, 19], as their structural attributes directly influence fluctuations in the popularity of public opinion [20, 21]. Therefore, indicator factors to evaluate the popularity of public opinion should be considered from the perspective of social circles. In this study, we construct an input feature system for public opinion popularity prediction using structural attributes, social attributes, and public opinion change variables of social circles.

The information entropy of the social circle can serve as an auxiliary indicator for characterizing the popularity of public opinion discussions within a specific time window. It reflects the dispersion of posting or interaction distributions within a group at a specific moment, thereby helping describe the complexity of public opinion discussions [22]. The structural attributes of social circles mainly describe the size and internal structural characteristics of cliques, including the number of members, the strength of the connection and the key characteristics of the topology of the network [23, 24]. The social attributes of social networks involve the level of activity of the user group on social media platforms. It includes the breadth of user information acquisition, the number of users that may be affected, and the level of enthusiasm to participate in topic discussions [25, 26]. The degree of change in public opinion serves as an indicator to evaluate the characteristics of performance and changes in the popularity of public opinion at a moment, revealing the development trend of public opinion events in social circles, such as the gradient and the degree of evolution change [27]. By measuring the dynamic changes and evolution of public opinion events in social networks, it is helpful to gain insight into the trajectory of public opinion events and provide important reference for further predicting and intervening in the development of public opinion events.

2.2 Social Network Construction Model

Large-scale influence is not primarily driven by influential individuals, but rather by a substantial number of individuals who are easily influenced [28]. Centrality metrics, including degree, k-shell, and eigenvalue centrality, can pinpoint the most influential nodes within a network [29, 30]. However, they are infrequently both useful and precise in quantifying the propagation capabilities of the vast majority of nodes that exert minimal influence. To better elucidate the propagation prowess of all network nodes, particularly in networks with excessively large adjacency matrices suitable for dynamic modeling, consider the distribution of diffusion potential generated by each node [31, 32]. The adoption of a new idea by an individual is frequently influenced by the decisions and attitudes of others within their social circle. Interaction probability models can be used to simulate reciprocal influence between individuals, predict the influence of nodes within the network, and facilitate more effective information dissemination [52].

Mathematical sociological models offer a quantitative framework for describing social structure and interpersonal interactions, providing a tool to analyze the dissemination of information in social networks [33]. By adopting the concept of interacting particle systems from physics, individuals within a network can be treated as interacting particles. By simulating the motion and behavior of these particles, the evolution of the entire system can be predicted, offering insights into the dynamics of social networks.

The graph thermodynamic theory introduces a graph theoretic framework for analyzing thermodynamic systems by employing process graph models to characterize interaction dynamics and interfacial connections during thermodynamic processes [34]. This modeling approach enables a concise representation of the system interconnectivity across diverse fluxes such as energy and popularity, while systematically integrating various thermodynamic work components, popularity transfer mechanisms, and their coupled interactions. The temperature between networks at consecutive time points can be directly derived from mathematically defined internal energy, where the internal energy is calculated as the ratio of the entropy variation to the energy variation. Crucially, all the thermodynamic variables in this framework are computable through basic network statistics, including network order and degree distributions [35].

Current network analysis predominantly adopts a static framework for studying dynamic systems by decomposing temporal networks into discrete snapshots and applying conventional structural metrics, such as clustering coefficients and degree correlations, to individual time slices. Although empirical studies demonstrate the effectiveness of this snapshot-based methodology in tracking specific structural evolution patterns, it fundamentally disregards intertemporal structural dependencies between consecutive steps, particularly the coupling mechanisms governing node degree variations and edge dynamics. This temporal decoupling limits the ability to holistically characterize the intrinsic evolutionary drivers underlying network transformations [36].

2.3 Public Opinion Prediction Method

Current popularity of public opinion prediction models rely on sentiment analysis and topic keyword techniques to extract text sentiment polarity or topic words frequency through LSTM or BERT [37, 50]. However, these methods have limitations as they only focus on emotional tendencies at the content level and ignore the critical role of network structure stability. Nodes update their representations through their adjacent, and they usually give the same weight to all neighboring nodes when they in traditional graph convolutional networks. In practice, different adjacent nodes may have different importance or influence on the target node.

The method based on complex network indicators identifies structural anomalies through node centrality and modularity detection, constructing a linear prediction model [39]. However, empirical research has shown that a single structural indicator has insufficient explanatory power for propagation volatility, and a multidimensional quantitative correlation between structural entropy, content divergence, and propagation speed has not been established. This makes it difficult for existing systems to distinguish between early signals of normal information diffusion and potential public opinion risks.

Time series graph modeling based on the attention mechanism optimizes node embedding through time-aware weights [40, 41, 51], but ignores the nonlinear characteristics of the evolution of the hierarchical structure. When a closed-loop interaction is formed between nodes, the collaborative changes in connection weights will trigger a step change in propagation patterns. For example, changes in user interaction frequency in social networks can synchronously alter the structural roles of nodes, transforming them from ordinary users to key communicators [42]. However, traditional models treat relationship types as fixed attributes and cannot capture the reshaping effect of dynamic identity transitions on communication paths.

Human-induced bias in hyperparameter optimization presents another critical issue in methodological design [43]. Conventional approaches, including grid search and random search, demonstrate limited efficiency when tuning graph neural networks (GNNs), while their reliance on researchers' empirical parameter range specification increases susceptibility to local optima [44]. Bayesian optimization addresses these limitations by constructing probabilistic models to characterize the hyperparameter-performance relationship, thereby allowing more efficient exploration of complex parameter spaces [48]. This study incorporates Bayesian optimization into the model framework by using a Gaussian-process surrogate to search for effective hyperparameter configurations, while the trainable GNN weights are optimized through gradient-based training. The proposed methodology systematically reduces the

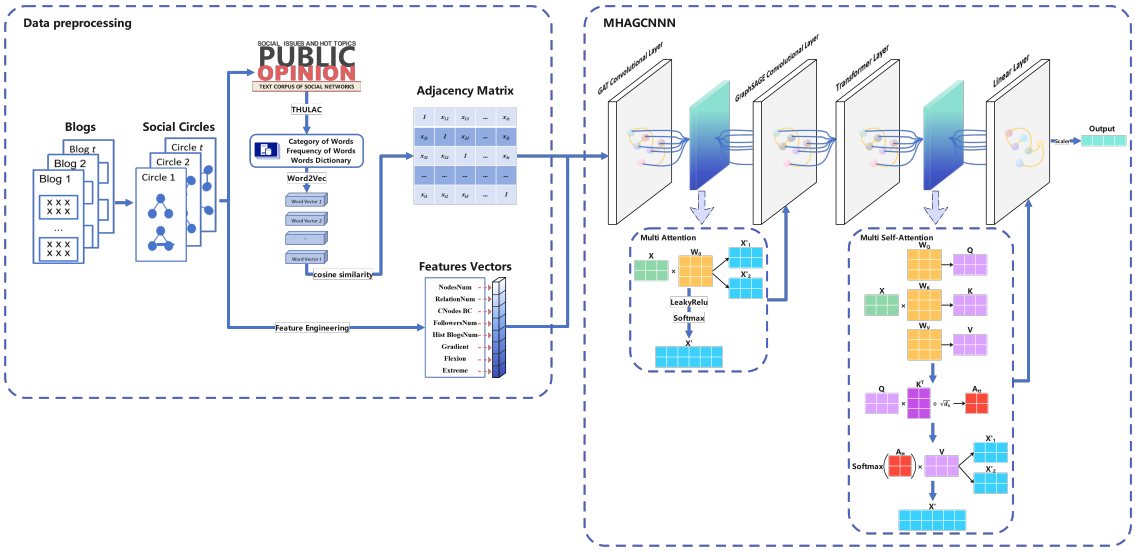


Fig. 1 Research model.

artificial bias introduced by manual intervention through its automated optimization mechanism.

3 Model Design and Algorithm Implementation

3.1 Research Model

This study proposes a social media data analysis framework that performs end-to-end modeling, spanning from raw data to predictive models, using a multi-stage processing workflow. The functions and calculation procedures of each module are elaborated as follows.

The first step is to perform data preprocessing and feature engineering. We build a custom dictionary and use the THULAC word segmentation tool to perform part of speech tagging and word frequency statistics on text data, generating domain specific dictionaries that can effectively improve the domain adaptability of subsequent word segmentation models. The word vector training adopts an improved Skip-Gram architecture, combined with precleaning to remove numbers and special characters, and stop word filtering, to train a word vector model. The generated domain-specific dictionaries are then loaded using the Pkuseg word segmentation tool for semantic feature extraction.

The second step extracts and models social network features. The social relation graph structure is derived from user interactions, which provide data to construct a dynamic adjacency matrix and compute multidimensional features. The node-level topological input variable used in X is betweenness centrality, while information entropy is computed separately as a time-window-level auxiliary statistic to characterize the diversity of posting or interaction distributions. The temporal characteristics include accumulating the number of nodes and relationships within a time window and calculating first-order differences (gradients), curvatures (inflection points), and extremum values. The specific meanings of each feature are detailed in Table 1. The generation of the node feature matrix and adjacency matrix follows two parallel paths: the node feature matrix X is constructed from the structural, social, and temporal variables listed in Table 1, whereas text embeddings are transformed into fixed-dimensional semantic representations and used to compute semantic similarity for adjacency construction. Specifically, PCA is employed when the initial semantic embedding dimension is higher than the target dimension, whereas random projection is used when dimension alignment is required for lower-dimensional embeddings.

The third step is the modeling and optimization of the graph neural network. Adjacency matrix post-processing aggregates edge

weights within an 8-hour time window, filters low-correlation edges, and adds self-loop edges to generate a sparse edge index. This significantly reduces computational complexity. To predict public opinion popularity, we introduce the MHAGCNN algorithm, which propagates multidimensional node features through the graph structure. This approach uses Bayesian hyperparameter optimization to determine optimal configurations. The detailed architecture and computational workflow of MHAGCNN are delineated in Algorithm 1. The comprehensive data analysis process model developed in this study is visually presented in Fig. 1.

3.2 Text Analysis Method

The text analysis framework proposed in this study implements a statistical language model-based approach to extracting feature characteristics of the word frequency, comprising three core phases, data preprocessing, frequency statistics, and lexicon generation. The formal mathematical definitions and algorithmic implementations are specified as follows.

Given a text corpus, we define the full document collection as $D_{\text{raw}} = \{d_1, d_2, \ldots, d_{N_{\text{doc}}}\}$, where N_{doc} denotes the number of valid non-null documents. Each document d_i consists of a character sequence $d_i = \{c_1, c_2, \ldots, c_{L_i}\}$, with L_i being the length of the i-th document. The subset used for lexicon initialization is defined as

$$D_{\text{lex}} = \{d_i\}_{i=1}^{N_{\text{lex}}}, \quad N_{\text{lex}} = \min(100, |D_{\text{raw}}|). \quad (1)$$

The lexicon is initialized from D_{lex}, whereas all valid comments in D_{raw} are used for feature extraction and model training.

Given a pretrained segmentation model with parameters ϕ, the segmentation process can be written as

$$Segment_{\phi}(d_i) = \{(w_{ij}, p_{ij})\}_{j=1}^{M_i}, \quad (2)$$

where w_{ij} and p_{ij} denote the j-th segmented word and its POS tag in document d_i, respectively.

Let Σ denote the Chinese character set, Σ^* denote the set of possible word strings, and P denote the POS tag set. In the corpus-level frequency statistic, for any lexical item $(w, p) \in \Sigma^* \times P$, the joint word-POS frequency metric is calculated as:

$$f(w, p) = \sum_{i=1}^{N_{\text{lex}}} \sum_{j=1}^{M_i} \mathbb{I}(w_{ij} = w \wedge p_{ij} = p). \quad (3)$$

where the indicator function $\mathbb{I}(\cdot)$ satisfies:

$$\mathbb{I}(\text{condition}) = \begin{cases} 1, & \text{if condition holds} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

The custom lexicon is sorted according to the following key:

$$\text{key}((w, p)) = (-f(w, p), w, p). \quad (5)$$

This ordering prioritizes high-frequency terms and uses lexicographical order to break ties.

3.3 Feature Engineering and the Adjacency Matrix

By conducting a systematic review and synthesis of the existing literature, this study identifies frequently used or validated characteristic variables in related research. We analyze their relevance to the dynamics of public opinion and address potential gaps or limitations. The feature variables employed in our model are proposed and refined through this process.

The feature engineering pipeline includes:

1. **Graph structure features**: Based on user interaction relationships, constructing a dynamic adjacency table and calculating multidimensional features
2. **Topological and auxiliary statistics**: Betweenness centrality is included in the node feature matrix X, while information entropy is computed as a time-window-level auxiliary statistic for temporal diversity analysis.
3. **Temporal characteristics**: Accumulating node and relationship counts within time windows, with calculations of:
   - First-order differences (gradients)
   - Curvatures (inflection points)
   - Extremum values

Table 1 Feature variables

Conceptualization	Features	Meaning and Explanation
Social Circle Structural Attributes	Event Impact Degree (NodesNum)	Total number of node users participating in discussions on public opinion events within the social circle network.
	Event Participation Degree (RelationNum)	Number of forwarding relationships established between node users.
	Information Control Power (CNodes BC)	Nodes located on multiple shortest paths between other nodes have greater betweenness centrality, reflecting stronger resource control capabilities and the ability to attract wider user discussion.
Social Circle Social Attributes	Follow Counts of Social Circle (FollowersNum)	Number of users followed by node users within the Social Circle, indicating the breadth of information sources for users within the Social Circle.
	Historical Tweets Counts of Social Circle (Hist BlogsNum)	Number of tweets posted by node users within the Social Circle in the past, indicating their enthusiasm for participating in discussions on public opinion event topics.
Social Circle Public Opinion Change Degree	Change Gradient of Public Opinion (Gradient)	First-order change rate of the public opinion popularity curve within the historical observation window, reflecting the instantaneous change in discussion intensity before prediction.
	Change Flection of Public Opinion (Flexion)	Curvature or inflection-related change of the public opinion popularity curve within the historical observation window, describing the intensity of trend changes before prediction.
	Extreme Popularity of Public Opinion (Extreme)	Maximum observed popularity value within the historical observation window, used to characterize the prior development scale of a public opinion event before prediction.

For each node i with text content C_i, the semantic text embedding $\mathbf{z}_i$ is computed as:

$$\mathbf{z}_i = \frac{1}{|C_i|} \sum_{w_k \in C_i} \mathbf{v}_k, \tag{6}$$

where $\mathbf{v}_k$ is the Word2Vec embedding of word w_k. The GNN input feature vector is denoted as $\mathbf{x}_i \in \mathbb{R}^F$, where $F = 8$ in this study. Specifically,

$$\mathbf{x}_i = [\mathit{NodesNum}_i,\ \mathit{RelationNum}_i,\ \mathit{CNodesBC}_i,\ \mathit{FollowersNum}_i,\ \mathit{HistBlogsNum}_i,\ \mathit{Gradient}_i,\ \mathit{Flexion}_i,\ \mathit{Extreme}_i]^T. \tag{7}$$

All variables in $\mathbf{x}_i$ are computed from the historical observation window before the prediction target is measured. The full feature matrix is $X = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T \in \mathbb{R}^{N \times F}$. The semantic representation $\tilde{\mathbf{z}}_i$ and the entropy statistic H_t are not concatenated into X; $\tilde{\mathbf{z}}_i$ is used for semantic similarity in adjacency construction, whereas H_t is used as an auxiliary statistic for temporal diversity analysis.

The semantic embedding vectors are transformed into a fixed-dimensional representation:

$$\tilde{\mathbf{z}}_i = \begin{cases} \mathbf{P}_{\mathrm{PCA}} \mathbf{z}_i, & d_z > d_0, \\ \mathbf{P}_{\mathrm{RP}} \mathbf{z}_i, & d_z \le d_0, \end{cases} \tag{8}$$

where $\mathbf{z}_i \in \mathbb{R}^{d_z}$ denotes the initial text embedding of node i, d_0 is the target dimension, $\mathbf{P}_{\mathrm{PCA}} \in \mathbb{R}^{d_0 \times d_z}$ denotes the PCA projection matrix, and $\mathbf{P}_{\mathrm{RP}} \in \mathbb{R}^{d_0 \times d_z}$ denotes the random projection matrix. Accordingly, $\tilde{\mathbf{z}}_i \in \mathbb{R}^{d_0}$. Let $A = [a_{ij}] \in \mathbb{R}^{N \times N}$ denote the weighted adjacency matrix. The

normalized interaction strength is computed as

$$\bar{r}_{ij} = \frac{r_{ij}}{\max_{u,v} r_{uv} + \epsilon_r}, \tag{9}$$

where r_{ij} denotes the observed forwarding or reposting frequency between users i and j, and ϵ_r is a small constant for numerical stability. The weighted adjacency matrix combines interaction strength and semantic similarity:

$$a_{ij} = \lambda \bar{r}_{ij} + (1-\lambda)\max(0, \cos(\tilde{\mathbf{z}}_i, \tilde{\mathbf{z}}_j)), \tag{10}$$

where $\bar{r}_{ij}$ denotes the normalized interaction strength between users i and j, $\cos(\tilde{\mathbf{z}}_i, \tilde{\mathbf{z}}_j)$ denotes semantic similarity based on the fixed-dimensional text embeddings, and $\lambda \in [0,1]$ balances structural and semantic information.

Self-loops are added and the matrix is symmetrically normalized:

$$\tilde{A} = D^{-\frac{1}{2}}(A + \mathbf{I}_N)D^{-\frac{1}{2}}, \tag{11}$$

where D is the degree matrix of $A + \mathbf{I}_N$, and $\mathbf{I}_N$ denotes the $N \times N$ identity matrix.

For temporal aggregation, posts and interactions are grouped into fixed 8-hour windows. Within each window, structural, social, temporal, and semantic features are computed and aligned with the corresponding user nodes. The semantic embeddings $\tilde{\mathbf{z}}_i$ are used to calculate semantic similarity, whereas the final GNN input X consists of the $F = 8$ variables listed in Table 1. The sparse edge set is obtained from the weighted adjacency matrix by thresholding:

$$\mathcal{E} = \{(i,j) \mid \tilde{a}_{ij} > \tau,\ i \neq j\} \cup \{(i,i) \mid i = 1, \ldots, N\}, \tag{12}$$

where $\tilde{a}_{ij}$ is the (i,j)-th element of $\tilde{A}$, and τ is a filtering threshold for low-correlation edges. The resulting edge set $\mathcal{E}$ is then represented by $edge_{index} \in \mathbb{N}^{2\times|\mathcal{E}|}$ for graph neural network training. If the 8-hour aggregation produces $T = 382$ temporal windows in the empirical dataset, this number refers to the number of temporal snapshots rather than the number of graph nodes.

3.4 Construction of Convolutional Layers

In traditional graph convolutional networks, nodes often assign uniform weights to all adjacent nodes during representation updating, which fails to capture the varying degrees of importance of neighboring nodes in real-world scenarios [46]. To address this limitation, we propose the MHAGCNN architecture, which employs a hierarchical design with residual connections to integrate GAT, GraphSAGE, and TransformerConv operators. The core GAT layer leverages a fusion attention mechanism to dynamically weight neighbors based on their relevance, enabling the model to simultaneously process node features and edge connection information. By emphasizing the optimization of feature propagation and attention mechanisms, MHAGCNN is specifically designed to capture hierarchical features and nuanced node dependencies that traditional methods overlook, thereby enhancing the modeling capability for complex graph data and improving prediction accuracy. MHAGCNN consists of four sequential stages. The first stage involves the construction of multi-head GAT layers. The input feature matrix is $X \in \mathbb{R}^{N\times F}$, where N denotes the number of graph nodes and $F = 8$ denotes the number of input variables listed in Table 1. Let $\mathbf{H}_1^{(0)} = X$, where the i-th row $\mathbf{h}_{1,i}^{(0)} = \mathbf{x}_i$ denotes the initial feature vector of node i. For the k-th attention head in the l-th GAT layer, the unnormalized attention score is computed as

$$e_{ij}^{(l,k)} = \text{LeakyReLU}\left(\mathbf{a}^{(l,k)T}[\mathbf{W}^{(l,k)}\mathbf{h}_{1,i}^{(l)} \| \mathbf{W}^{(l,k)}\mathbf{h}_{1,j}^{(l)}]\right), \tag{13}$$

where $\mathbf{W}^{(l,k)}$ is a learnable transformation matrix, $\mathbf{a}^{(l,k)}$ is a learnable attention vector, and $\|$ denotes vector concatenation. The normalized attention coefficient is

$$\alpha_{ij}^{(l,k)} = \frac{\exp(e_{ij}^{(l,k)})}{\sum_{u\in\mathcal{N}^+(i)} \exp(e_{iu}^{(l,k)})}, \tag{14}$$

where $\mathcal{N}^+(i) = \mathcal{N}(i) \cup \{i\}$ includes the self-loop of node i. The output of the multi-head GAT layer is

$$\mathbf{h}_{1,i}^{(l+1)} = \text{ELU}\left(\text{BN}\left(\Big\|_{k=1}^{K_{\text{gat}}} \sum_{j\in\mathcal{N}^+(i)} \alpha_{ij}^{(l,k)}\mathbf{W}^{(l,k)}\mathbf{h}_{1,j}^{(l)}\right)\right), \tag{15}$$

and the node representation matrix is

$$\mathbf{H}_1^{(l+1)} = [\mathbf{h}_{1,1}^{(l+1)}, \ldots, \mathbf{h}_{1,N}^{(l+1)}]^T. \tag{16}$$

The second stage is the construction of the GraphSAGE aggregation layer. After the stacked GAT layers, the output representation of node i is denoted as $\mathbf{h}_{1,i}^{(L)}$, and the corresponding node representation matrix is denoted as $\mathbf{H}_1^{(L)} = [\mathbf{h}_{1,1}^{(L)}, \mathbf{h}_{1,2}^{(L)}, \ldots, \mathbf{h}_{1,N}^{(L)}]^T$. To further aggregate local neighborhood information, a GraphSAGE layer is applied:

$$\mathbf{h}_{2,i} = \sigma\left(\mathbf{W}_{\text{sage}}\left[\mathbf{h}_{1,i}^{(L)} \| \operatorname{AGG}\left(\{\mathbf{h}_{1,j}^{(L)} : j \in \mathcal{N}^+(i)\}\right)\right]\right), \tag{17}$$

where $\operatorname{AGG}(\cdot)$ denotes a neighborhood aggregation function, $\|$ denotes vector concatenation, $\mathbf{W}_{\text{sage}}$ is a learnable weight matrix, and $\sigma(\cdot)$ denotes the ELU activation function. The output matrix of the GraphSAGE layer is

$$\mathbf{H}_2 = [\mathbf{h}_{2,1}, \mathbf{h}_{2,2}, \ldots, \mathbf{h}_{2,N}]^T. \tag{18}$$

The third stage uses TransformerConv to model higher-order dependencies over graph-defined neighborhoods. Unlike a standard Transformer that computes attention over all nodes, TransformerConv restricts attention to the graph neighborhood specified by $edge_{index}$. For node i, $\mathcal{N}^+(i)$ denotes the self-loop-augmented neighborhood defined above. For the k-th TransformerConv head, the query, key, and value vectors are computed from the GraphSAGE output $\mathbf{H}_2$ as follows:

$$\mathbf{q}_i^{(k)} = \mathbf{W}_Q^{(k)}\mathbf{h}_{2,i}, \quad \boldsymbol{\kappa}_j^{(k)} = \mathbf{W}_K^{(k)}\mathbf{h}_{2,j}, \quad \mathbf{v}_j^{(k)} = \mathbf{W}_V^{(k)}\mathbf{h}_{2,j}, \tag{19}$$

where $\mathbf{h}_{2,i}$ is the GraphSAGE representation of node i, and $\mathbf{W}_Q^{(k)}$, $\mathbf{W}_K^{(k)}$, and $\mathbf{W}_V^{(k)}$ are learnable projection matrices for the k-th attention head. Here, $\mathbf{q}_i^{(k)}$, $\boldsymbol{\kappa}_j^{(k)}$, and $\mathbf{v}_j^{(k)}$ denote the query, key, and value vectors, respectively.

The attention coefficient from node j to node i is computed only over the graph-defined neighborhood:

$$\beta_{ij}^{(k)} = \frac{\exp\left(\frac{(\mathbf{q}_i^{(k)})^T\boldsymbol{\kappa}_j^{(k)}}{\sqrt{d_k}}\right)}{\sum_{u\in\mathcal{N}^+(i)}\exp\left(\frac{(\mathbf{q}_i^{(k)})^T\boldsymbol{\kappa}_u^{(k)}}{\sqrt{d_k}}\right)}, \quad j \in \mathcal{N}^+(i), \tag{20}$$

where d_k is the dimensionality of each attention head.

The output representation of node i in the k-th TransformerConv head is obtained by weighted neighborhood aggregation:

$$\mathbf{h}_{3,i}^{(k)} = \sum_{j\in\mathcal{N}^+(i)} \beta_{ij}^{(k)}\mathbf{v}_j^{(k)}. \tag{21}$$

The node representations of all nodes in the k-th head form the matrix

$$\mathbf{H}_3^{(k)} = [\mathbf{h}_{3,1}^{(k)}, \mathbf{h}_{3,2}^{(k)}, \ldots, \mathbf{h}_{3,N}^{(k)}]^T. \tag{22}$$

Finally, the multi-head TransformerConv outputs are concatenated as

$$\mathbf{H}_3 = \Big\|_{k=1}^{K_{\text{trans}}}\mathbf{H}_3^{(k)}. \tag{23}$$

Thus, $\mathbf{H}_3 \in \mathbb{R}^{N\times d_{\text{in}}}$, where $d_{\text{in}} = K_{\text{trans}}d_k$ when all attention heads have the same dimensionality d_k. The vector $\mathbf{h}_{3,i} \in \mathbb{R}^{d_{\text{in}}}$ denotes the i-th row of $\mathbf{H}_3$, namely the final TransformerConv representation of node i. The fourth stage is the final regression layer, which includes a linear projection to generate the final prediction. Here, $\mathbf{w}_{\text{final}} \in \mathbb{R}^{d_{\text{in}}}$ and $b_{\text{final}} \in \mathbb{R}$ are the learnable parameters of the regression layer, where d_{in} corresponds to the concatenated feature dimension of the TransformerConv output.

The supervised target y_i denotes the normalized public opinion popularity associated with node i in the prediction window. It is derived from observed popularity-related engagement signals, such as forwarding or reposting activity, and is not included in the input feature vector $\mathbf{x}_i$. For each node i, the final prediction is computed as

$$\hat{y}_i = \mathbf{w}_{\text{final}}^T\mathbf{h}_{3,i} + b_{\text{final}}. \tag{24}$$

Equivalently, for all nodes, the prediction vector can be written as

$$\hat{\mathbf{y}}^T = \mathbf{w}_{\text{final}}^T\mathbf{H}_3^T + b_{\text{final}}\mathbf{1}_N^T. \tag{25}$$

This is equivalent to the column-vector form

$$\hat{\mathbf{y}} = \mathbf{H}_3\mathbf{w}_{\text{final}} + b_{\text{final}}\mathbf{1}_N. \tag{26}$$

During training, the model parameters are optimized to minimize the Mean Squared Error (MSE) loss function. This loss function computes the average squared discrepancies between the

ground truth target values y_i and the predicted values by the model $\hat{y}_i$ for all nodes i within the training set. The loss function is defined as

$$\mathcal{L} = \frac{1}{|\mathcal{V}_{\text{train}}|} \sum_{i \in \mathcal{V}_{\text{train}}} (y_i - \hat{y}_i)^2. \tag{27}$$

We employ gradient clipping to prevent vanishing/exploding gradients and use a ReduceLROnPlateau scheduler to adjust the learning rate dynamically. Early stopping is applied to avoid overfitting, terminating training if validation loss does not improve for 30 epochs.

3.5 Optimization and Training Strategies

In the Optimization and Training Strategies component of our model design, the incorporation of Bayesian hyperparameter optimization is crucial to enhancing the predictive precision of MHAGCNN by systematically navigating the complex hyperparameter landscape inherent in its architecture [49]. Let $\mathbf{p}$ denote the hyperparameter vector and $g(\mathbf{p})$ denote the validation performance. Bayesian optimization models $g(\mathbf{p})$ using a Gaussian-process surrogate:

$$g(\mathbf{p}) \sim \mathcal{GP}\left(\mu(\mathbf{p}), k(\mathbf{p}, \mathbf{p}')\right). \tag{28}$$

This approach is particularly critical for MHAGCNN due to the model's integration of various graph-based operators, including GAT, GraphSAGE, and TransformerConv, which introduce a multifaceted hyperparameter space encompassing attention heads, aggregation strategies, learning rates, and regularization parameters. Manual tuning of such a complex landscape is inefficient and often results in suboptimal configurations, whereas Bayesian optimization balances exploration of untested regions and exploitation of promising areas, enabling the discovery of hyperparameter combinations that optimally calibrate feature propagation and attention mechanisms to align with the model's design objectives [38].

The critical hyperparameters of MHAGCNN are optimized using Bayesian optimization, and the search space includes the number of GAT layers, the number of attention heads, dropout, learning rate, weight decay, and gradient clipping

Algorithm 1 Bayesian Optimization of MHAGCNN

Require: Raw feature matrix X_{raw}, Adjacency matrix A, Target values y_{raw}
Ensure: Optimized model M^*, Test predictions $\hat{y}_{test}$, Evaluation metric R^2

// Data Preprocessing
Preprocess X_{raw} and y_{raw} to obtain X_{scaled} and y_{scaled}
Normalize A to obtain $\bar{A}$, threshold $\bar{A}$ to obtain $\mathcal{E}$, and construct $edge_{index}$
// Basic Data Construction
$base_data \leftarrow Data\begin{pmatrix} x = X_{scaled} \\ edge_{index} = edge_{index} \\ y = y_{scaled} \end{pmatrix}$
// Optimization Setup
Initialize parameter space $\mathcal{P}$ and optimization tracker
$S \leftarrow$ GaussianProcess() ▷ Bayesian optimization surrogate
$p^* \leftarrow \emptyset$, $best_score \leftarrow -\infty$
for $t = 1$ **to** T_{opt} **do** ▷ Optimization iterations
  $p_t \leftarrow S.suggest\ parameters()$ ▷ Propose parameters
  // Data Splitting
  $data_t \leftarrow create_split(base_data)$
  // Model Training
  $M_t \leftarrow MHAGCNN(params = p_t)$
  Initialize optimizer with parameters from p_t
  best $R^2 \leftarrow -\infty$
  for $epoch = 1$ **to** E **do** ▷ Training loop
    Perform forward pass and compute loss
    Backpropagate and update model parameters
    $r_t^{epoch} \leftarrow$ evaluate_validation_performance(M_t)
    best $R^2 \leftarrow \max(\text{best } R^2, r_t^{epoch})$
  end for
  // Update Optimization State
  if best$R^2 > best_score$ **then**
    $p^* \leftarrow p_t$, $best_score \leftarrow$ best R^2
  end if
  Update tracker and surrogate model S
end for
// Final Model Training
final_data $\leftarrow$ create_final_split(base_data)
$M^* \leftarrow$ MHAGCNN(params $= p^*$)
Train M^* with early stopping on final_data
// Evaluation
$\hat{y}_{\text{test}} \leftarrow M^*$(final_data)[test_mask]
Compute evaluation metrics R^2
return $M^*, \hat{y}_{\text{test}}, R^2$

threshold. We also implemented a ReduceLROnPlateau scheduler that reduces the learning rate when the validation loss stops improving. To prevent overfitting, training is terminated if the validation loss does not improve for 30 consecutive epochs.

$$\eta_{e+1} = \begin{cases} \rho\eta_e, & \text{if } \mathcal{L}_{\text{val}} \text{ does not improve for } P \text{ epochs}, \\ \eta_e, & \text{otherwise}, \end{cases} \tag{29}$$

Here, e indexes training epochs, whereas t in the AdamW update denotes parameter-update steps. And, $\rho \in (0, 1)$ is the learning-rate decay factor, and P is the patience parameter.

The optimizer hyperparameters include the learning rate and weight decay. During the training process, gradient calculation and parameter updates are performed at each iteration. Eq. (30) describes the use of parameter update rules for model parameters in each training step, corresponding to the core steps of the Adam Weight Decay optimization algorithm (AdamW), where θ_t denotes the model parameters at the t-th iteration, $\hat{m}_t$ is the bias-corrected first-moment estimate, $\hat{v}_t$ is the bias-corrected second-moment estimate, λ is the weight-decay coefficient, and ϵ is a small constant for numerical stability.

$$\theta_{t+1} = (1 - \eta_t \lambda)\theta_t - \eta_t \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}, \tag{30}$$

After each calculation of the backpropagation gradient, the model parameters are updated according to this formula to minimize the loss function $\mathcal{L}$. This mechanism significantly improves the convergence speed and generalization ability of the model on complex graph structured data. The meanings of some parameter indicators of the model are shown in Table 2.

4 Experimental Analysis

4.1 Data Preprocessing

In this study, we selected social media data from the Weibo platform, which is one of the most popular social media platforms in China. The dataset contains 112,209 Weibo comment records and corresponding user attribute information related to hot topics and public opinion events. In the constructed graph, users are represented as nodes, and observed interactions are represented as edges. The selection of this data set was based on its relevance to our research objectives and its ability to reflect real-world social interactions and information diffusion patterns.

The raw data was pre-processed through several key steps to ensure their quality and suitability for subsequent analysis. First, we performed text cleaning on user comments, including the removal of URLs, emojis, special characters, duplicate symbols, and invalid records, followed by Chinese word segmentation. We also removed stop words using a predefined list of Chinese stop words to reduce noise in the text data.

Next, we built a social network graph based on user interactions. The nodes in the graph represent users, and the edges represent forwarding or reposting relationships. The adjacency matrix of this graph was calculated and processed to capture the structural relationships between users.

Finally, to extract meaningful features from the text data, we employed word vectorization techniques. Specifically, THULAC and Pkuseg were used for word segmentation, and a Skip-Gram Word2Vec model was trained on the segmented corpus to generate word embeddings. These embeddings were used to represent user comments as numerical feature vectors.

4.2 Feature Extraction

We used a comprehensive feature engineering approach to capture both the structural and semantic characteristics of user nodes in the social media network. Structural features were derived from the network topology, including metrics such as betweenness centrality, which quantifies a node's influence by measuring how frequently it lies on the shortest paths between other nodes.

Text features were extracted from user-generated content. We used the Skip-Gram Word2Vec model trained on the segmented corpus to convert tokenized comments into numerical vectors, capturing semantic information. Each comment was represented as the average of its constituent word vectors. Temporal and activity-related features, such as historical tweet counts, gradient, flexion, and extreme popularity values, were computed to characterize user activity patterns and public opinion change dynamics. As an auxiliary statistic that is not included in the node feature matrix X, information entropy was used to assess the popularity of public opinion within each time window.

For a given time window t, information entropy is computed as

$$H_t = -\sum_{q=1}^{Q} p_{tq} \log p_{tq}, \tag{31}$$

where p_{tq} denotes the normalized posting or interaction frequency of unit q in time window t. A larger H_t indicates a more dispersed posting or interaction distribution within the time window, suggesting higher temporal diversity and

Table 2 MHAGCNN variables

Variable	Dimension	Description
X	$N \times F$	Node feature matrix.
A	$N \times N$	Weighted adjacency matrix.
$\tilde{A}$	$N \times N$	Normalized adjacency matrix with self-loops.
H_t	Scalar	Auxiliary information entropy statistic for time window t, not included in X.
$edge_{index}$	$2 \times \lvert\mathcal{E}\rvert$	Sparse edge index matrix.
$\mathbf{h}_{1,i}^{(l)}$	d_l	GAT representation of node i at layer l.
$\mathbf{h}_{2,i}$	d_{sage}	GraphSAGE representation of node i.
$\mathbf{h}_{3,i}$	d_{in}	TransformerConv representation of node i.
$\mathcal{N}(i)$	Set	Neighbor set of node i.
$\alpha_{ij}^{(l,k)}$	Scalar	Attention coefficient from node j to node i.
$\beta_{ij}^{(k)}$	Scalar	TransformerConv attention coefficient from node j to node i in head k.
$\mathbf{p}$	–	Hyperparameter vector.
θ	–	Trainable model parameters.
η_e, η_t	Scalar	Learning rate at epoch e and parameter-update step t, respectively.

greater complexity of public opinion dynamics. Statistical features were derived from time-series analysis of user activity. Gradient and flexion metrics were calculated to capture the rate of change and the curvature of the posting frequency over time, respectively. Extreme values were identified to represent peak periods of user engagement. These input features provide a multi-faceted representation of user behavior and public opinion change, while the auxiliary entropy statistic supports interpretation of temporal diversity.

The adjacency matrix, representing relationships between users, was constructed using a combination of interaction-based and semantic similarity measures. We first counted the number of forwarding or reposting interactions between each pair of users over a specified time window and normalized the resulting interaction strength. To incorporate semantic similarity, we computed the cosine similarity between the fixed-dimensional text embeddings of user comments. The final adjacency matrix was obtained by combining these two measures, allowing us to capture both the topological structure of the network and the semantic relatedness of the user content. Self-loops were added to ensure that each node had a connection to itself, and the matrix was symmetrically normalized to facilitate effective graph neural network processing. This approach ensured that the adjacency matrix comprehensively represented the complex relationships within the social media ecosystem.

4.3 Baseline Model and Bayesian Optimization

Fig. 2 presents two complementary subplots that collectively elucidate the training behavior and optimization strategies of the model. The left subplot tracks the progressive reduction of training (blue trajectory) and validation (orange trajectory) losses across successive epochs. Both curves initially manifest elevated values consistent with the initialization of the random parameters, followed by a steep decline during the initial training phases, indicative of rapid feature acquisition. While the training loss displays higher stochasticity attributable to mini batch gradient descent variations, the validation curve demonstrates smoother stabilization after epoch 10, with both metrics converging asymptotically toward minimal values. The persistent parallelism between these curves throughout training implies effective regularization and absence of pathological overfitting, as no substantial divergence emerges between training and validation performance metrics.

The right subplot chronicles the dynamic learning rate adaptation strategy, beginning with an elevated initial value that undergoes discrete stepwise reductions at predetermined intervals.

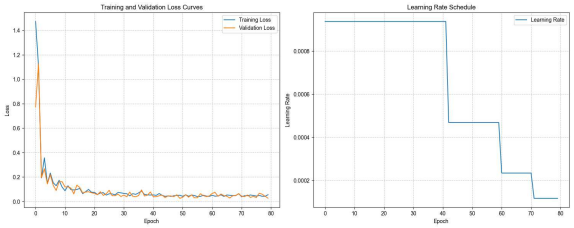


Fig. 2 Training and validation loss curves with learning-rate adaptation.

This pattern aligns with common scheduling paradigms such as ReduceLROnPlateau, where rate adjustments are triggered by plateau detection in validation loss. Deliberate modulation facilitates refined parameter updates during later training stages, mitigating oscillation risks while preventing premature convergence to suboptimal minima.

The synthesis of these observations reveals three critical performance characteristics: First, the monotonic descent of both loss curves confirms sustained learning capacity throughout the training regimen. Second, the maintained congruence between training and validation metrics substantiates the model's generalizability, with no evidence of over-parameterization compromising performance on unseen data. Third, the implemented learning rate scheduling proves instrumental in balancing exploration exploitation trade-offs, enabling stable navigation through loss landscapes.

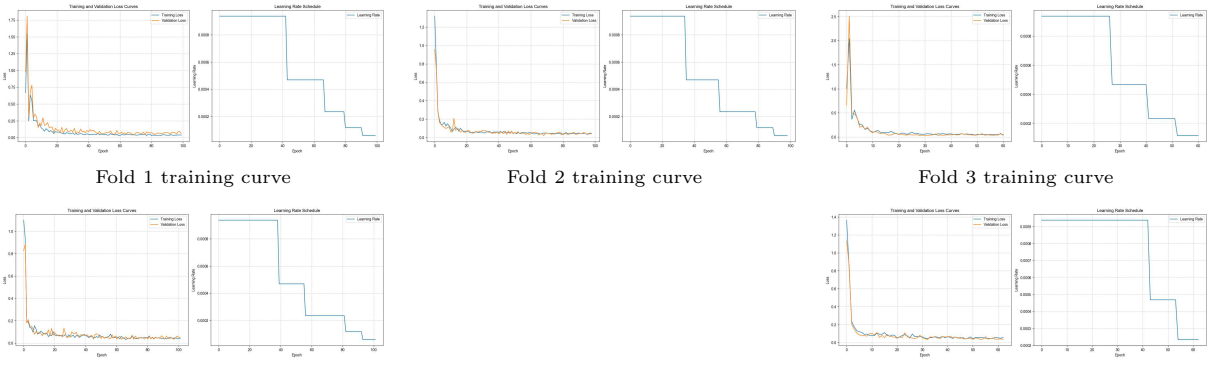
The 5-fold cross validation results demonstrate the robustness and generalization capability of the MHAGCNN model as shown in Fig. 3. The training process exhibited efficient convergence, with early stopping triggered between epochs 60 and 101, indicating optimal parameter tuning and prevention of overfitting across all folds, but performance metrics varied slightly among different folds. Fold 2 achieved the highest R^2 score (0.986) and lowest RMSE (0.223), while Fold 3 showed the lowest R^2 (0.932) and the highest RMSE (0.456). These fluctuations may be attributed to inherent heterogeneity of the data or varying complexity in the feature distributions between subsets.

The average metrics (MSE $= 0.124 \pm 0.061$, RMSE $= 0.340 \pm 0.091$, $R^2 = 0.965 \pm 0.021$) reflect strong overall predictive power, with R^2 exceeding 0.93 across all folds, indicating that the model explains more than 93% of the variance in each cross-validation fold. The relatively low standard deviations underscore the stability of the model, although the higher variance in Fold 3 suggests potential sensitivity to specific data patterns.

These results align with the model architecture design, which integrates GAT, GraphSAGE, and TransformerConv layers to capture structural, temporal, and semantic relationships in social media networks. The early stopping mechanism, learning rate scheduling, and hyperparameter optimization implemented in the code effectively balanced training efficiency and generalization. Minor discrepancies across folds highlight opportunities for refining feature engineering or augmenting data preprocessing to address subset-specific challenges.

Overall, the cross-validation results validate the model's efficacy in predicting public opinion popularity from structural, semantic, and temporal features.

Bayesian parameter optimization systematically explored the hyperparameter space to identify configurations that maximize model performance, as visualized in Figure 4. For *conv1_channels*, the optimization process identified an optimal value within the range of 32 to 128, balancing the prediction capacity of the model and computational efficiency to achieve the highest R^2 score. The *conv3_channels* parameter was optimized between 16 and 64, the optimal value ensuring effective feature extraction in the final convolutional layers. *Gat_heads*, controlling the number of attention heads in the GAT layers, was tuned between 2 and 8, with the optimal setting enhancing the model's ability to capture diverse relationships in the graph structure. The *trans_heads* parameter, adjusted between 0 (representing 4 heads) and 1 (representing 8 heads), optimized the attention mechanism in the TransformerConv layer, improving graph-context representation learning. *Dropout*, set between 0.1 and 0.5, was optimized to prevent overfitting, with the optimal value balancing regularization and model expressiveness. The *lr* was finely tuned within the range of 10^{-5} to 10^{-3}, ensuring stable convergence during training, while the *weight_decay*, optimized between 10^{-5} and 10^{-3}, regulated the complexity of the model to avoid overfitting. *Grad_clip*, set between 1.0 and 5.0, prevented gradient explosion during backpropagation, maintaining training



Fold 1 training curve

Fold 2 training curve

Fold 3 training curve

Fold 4 training curve

Fold 5 training curve

Fig. 3 Training curves for all five folds.

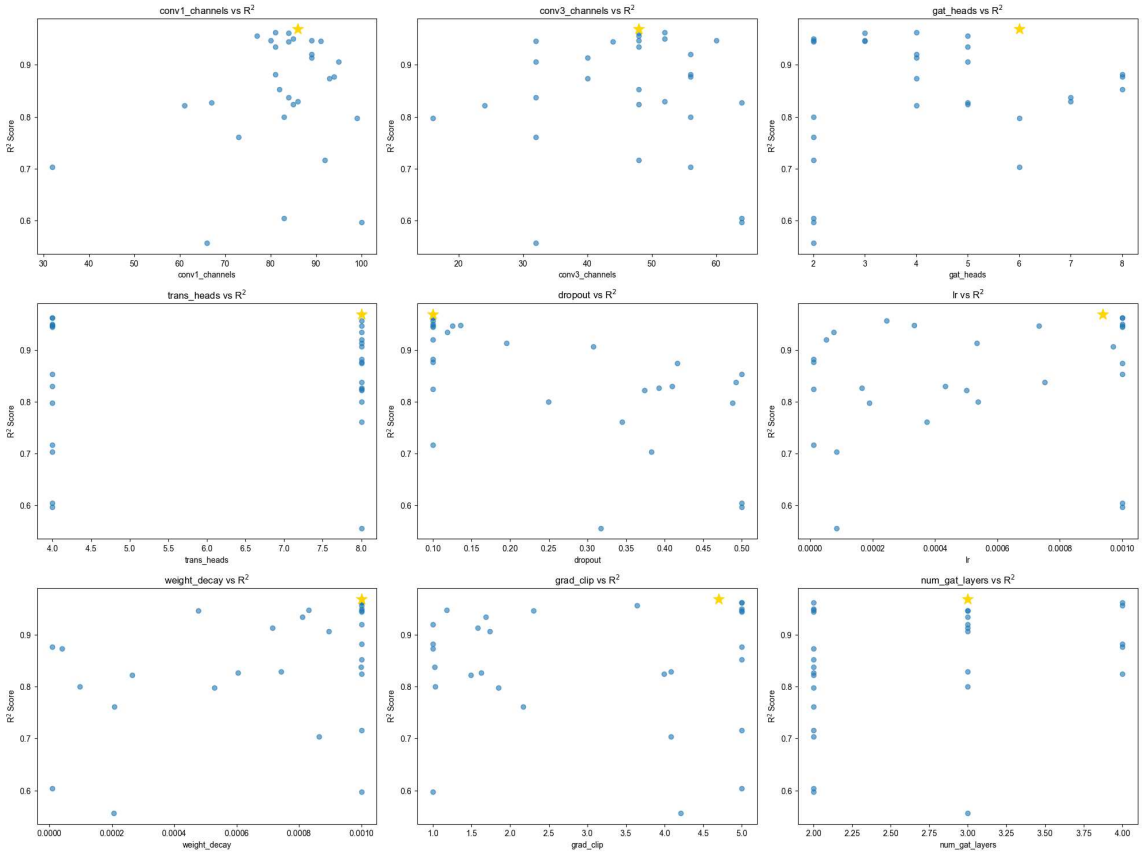


Fig. 4 Results of Bayesian parameter optimization.

stability. Finally, the *num_gat_layers* parameter, adjusted between 2 and 4, determined the depth of the GAT architecture, with the optimal value striking a balance between model depth and training efficiency. The optimized hyperparameters enabled the MHAGCNN model to achieve robust performance and generalization in the validation folds.

Collectively, these findings underscore the efficacy of the adopted training protocol, which demonstrates coordinated parameter optimization, robust generalization capacity, and strategic

management of learning rate, all hallmarks of a well-engineered machine learning system.

4.4 Model Comparison

The comparison of the model reveals that the MHAGCNN model achieves the best performance, as evidenced by its lowest MSE (0.129), RMSE (0.359) and highest R^2 (0.967). This superiority can be attributed to its unique combination of capabilities that address the limitations of other models. As shown in Table 3, traditional machine learning models such as XGBoost, SVM, and Random Forest lack the ability to leverage the spatial structure information inherent in graph-structured data, severely restricting their performance in capturing relational patterns. Neural network models such as BiLSTM, SampleCNN, and MLP can model nonlinear feature interactions, but they do not explicitly exploit graph-structured relational information. While the Transformer model incorporates an attention mechanism, it does not explicitly utilize graph-structured data, missing out on the structural insights that graph models can extract. In contrast, the MHAGCNN model effectively processes graph-structured data through graph convolutional layers, mining structural features from node connections to enhance prediction accuracy. In addition, its attention mechanisms allow it to assign adaptive weights to neighboring nodes and graph-context representations, prioritizing information that has a greater impact on predictions. By integrating both the exploitation of spatial structure and the emphasis on important features via attention, MHAGCNN surpasses other models that lack one or both of these crucial components, thereby demonstrating superior performance in handling complex, structured data.

Table 3 Results of model comparison

Model	MSE	RMSE	R^2
XGBoost	0.449	0.670	0.871
SVM	0.738	0.859	0.863
Random Forest	1.205	1.098	0.777
BiLSTM	1.609	1.268	0.702
SampleCNN	1.182	1.087	0.781
MLP	1.039	1.019	0.808
Transformer	0.284	0.533	0.918
MHAGCNN	**0.129**	**0.359**	**0.967**

The advantage of MHAGCNN arises from the complementary roles of its three graph learning components. The GAT layer assigns adaptive attention weights to neighboring nodes, enabling the model to distinguish their different contributions. The GraphSAGE layer further aggregates neighborhood information to enhance local structural representation. The TransformerConv layer captures higher-order graph-context dependencies through attention over graph-defined neighborhoods. Together, these components allow MHAGCNN to jointly exploit node features, local graph structure, and semantic-contextual relationships, thereby improving model expressiveness and predictive performance.

5 Conclusion and Discussion

This study presents a novel MHAGCNN model that integrates GAT, GraphSAGE, and TransformerConv layers to effectively capture structural, temporal, and semantic relationships in social media networks. Through comprehensive experiments, the model demonstrated superior performance over traditional machine learning methods and standalone neural networks, achieving an average R^2 score of 0.965 in five-fold cross validation and an R^2 score of 0.967 in the final baseline comparison. Bayesian optimization of hyperparameters further refined the predictive accuracy of the model, highlighting the importance of adjusting parameters to balance model complexity and generalization. The MHAGCNN variant, which combines graph convolutional layers with attention mechanisms, outperformed all baselines, underscoring the critical role of leveraging graph structure, neighborhood importance, and semantic-contextual dependencies. These results validate the effectiveness of integrating advanced graph neural network architectures to model complex social interactions.

The significance of deploying a Bayesian optimization strategy is that it mitigates human bias through automated tuning, ensuring that the architecture is rigorously adapted to the unique characteristics of graph-structured data, such as node heterogeneity and edge sparsity, thus strengthening the theoretical foundation of the model by anchoring its performance in data-driven optimal parameters [47]. In essence, Bayesian hyperparameter optimization transforms

MHAGCNN from a versatile yet generic framework into a tailored solution that maximizes predictive power by leveraging data-specific hyperparameter configurations, solidifying its efficacy in addressing the challenges of graph-structured analysis.

In theory, this study contributes to dynamic social network modeling by showing that structural, semantic, and temporal features can be jointly represented through a hybrid graph neural architecture. This study also contributes to the theoretical understanding of graph neural networks by demonstrating that incorporating multilayer graph convolutional architectures with attention mechanisms can significantly enhance the extraction of meaningful patterns from social media data. The study highlights that structural features derived from user interactions, when combined with semantic information through graph attention-based weighting, provide a more comprehensive representation of social dynamics than traditional methods. Furthermore, the hyperparameter optimization process offers insight into the sensitivity of GNN performance to architectural choices, informing future model design in similar domains.

In practice, the proposed model can support social media analysis tasks such as public opinion popularity prediction, information diffusion monitoring, and influential-user identification. By accurately capturing user relationships and content semantics, the model can assist policymakers, marketers, and researchers in making data-driven decisions. The efficient early stopping and learning rate scheduling strategies implemented in the model also contribute to practicality by reducing computational costs and training time, making the approach scalable for large-scale social media datasets.

Future work may focus on three directions. First, more diverse social media platforms and event types can be included to test the generalizability of the model. Second, explicit risk-level labels and early-warning metrics can be introduced to evaluate public opinion risk classification. Third, interpretable graph learning methods can be incorporated to explain how structural, semantic, and temporal factors contribute to prediction results.

Acknowledgements. Not applicable.

Declarations. **Funding** Not applicable.
Competing interests The authors declare that they have no competing interests.
Ethics approval Not applicable.
Consent to participate Not applicable.
Consent for publication Not applicable.
Data availability Due to platform terms and privacy considerations, the raw Weibo data are not publicly available. De-identified derived data supporting the main findings are available from the corresponding author upon reasonable request.
Code availability The code used for data preprocessing, graph construction, model training, and evaluation is available from the corresponding author upon reasonable request.
Author contribution Yongqing Yang: Conceptualization, Methodology, Data curation, Writing – original draft. Chenghao Fan: Formal analysis, Validation, Writing – review & editing. Linhan Ouyang: Supervision, Methodology, Writing – review & editing. All authors read and approved the final manuscript.

References

[1] A. Edelmann, T. Wolff, D. Montagne, and C. A. Bail, "Computational social science and sociology," *Annual Review of Sociology*, vol. 46, no. 1, pp. 61–81, 2020.

[2] P. Cairney, "Complexity theory in political science and public policy," *Political Studies Review*, vol. 10, no. 3, pp. 346–358, 2012.

[3] G. Deffuant, D. Neau, F. Amblard, and G. Weisbuch, "Mixing beliefs among interacting agents," *Advances in Complex Systems*, vol. 3, no. 01n04, pp. 87–98, 2000.

[4] V. D. Blondel, J. M. Hendrickx, and J. N. Tsitsiklis, "On Krause's multi-agent consensus model with state-dependent connectivity," *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2586–2597, 2009.

[5] K. Sznajd-Weron, J. Sznajd, and T. Weron, "A review on the Sznajd model—20 years after," *Physica A: Statistical Mechanics and its Applications*, vol. 565, p. 125537, 2021.

[6] N. Crokidakis, "Effects of mass media on opinion spreading in the Sznajd sociophysics model," *Physica A: Statistical Mechanics and its Applications*, vol. 391, no. 4, pp. 1729–1734, 2012.

[7] Y. Liu and S. Xu, "Detecting rumors through modeling information propagation networks in a social media environment," *IEEE Transactions on Computational Social Systems*, vol. 3, no. 2, pp. 46–62, 2016.

[8] F. Liu and M. Buss, "Optimal control for heterogeneous node-based information epidemics over social networks," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 3, pp. 1115–1126, 2020.

[9] A. De, I. Valera, N. Ganguly, S. Bhattacharya, and M. Gomez Rodriguez, "Learning and forecasting opinion dynamics in social networks," *Advances in Neural Information Processing Systems*, 2016, pp. 1–9.

[10] B. D. Anderson and M. Ye, "Recent advances in the modelling and analysis of opinion dynamics on influence networks," *International Journal of Automation and Computing*, vol. 16, no. 2, pp. 129–149, 2019.

[11] J. Denrell and G. Le Mens, "Information sampling, belief synchronization, and collective illusions," *Management Science*, vol. 63, no. 2, pp. 528–547, 2017.

[12] H. Nguyen, R. Calantone, and R. Krishnan, "Influence of social media emotional word of mouth on institutional investors' decisions and firm value," *Management Science*, vol. 66, no. 2, pp. 887–910, 2020.

[13] A. Kim and A. R. Dennis, "Says who? The effects of presentation format and source rating on fake news in social media," *MIS Quarterly*, vol. 43, no. 3, pp. 1025–1040, 2019.

[14] N. Zhong and D. A. Schweidel, "Capturing changes in social media content: A multiple latent changepoint topic model," *Marketing Science*, vol. 39, no. 4, pp. 827–846, 2020.

[15] A. Abbasi, Y. Zhou, S. Deng, and P. Zhang, "Text analytics to support sense-making in social media," *MIS Quarterly*, vol. 42, no. 2, pp. 427–464, 2018.

[16] Y. J. Lee, K. Hosanagar, and Y. Tan, "Do I follow my friends or the crowd? Information cascades in online movie ratings," *Management Science*, vol. 61, no. 9, pp. 2241–2258, 2015.

[17] P. L. Moravec, R. K. Minas, and A. R. Dennis, "Fake News on Social Media," *MIS Quarterly*, vol. 43, no. 4, pp. 1343–A13, 2019.

[18] J. E. Tropman, "Critical dimensions of community structure: A reexamination of the Hadden-Borgatta findings," *Urban Affairs Quarterly*, vol. 5, no. 2, pp. 215–232, 1969.

[19] R. J. Doolittle and D. MacDonald, "Communication and a sense of community in a metropolitan neighborhood: A factor analytic examination," *Communication Quarterly*, vol. 26, no. 3, pp. 2-7, 1978.

[20] C. Kadushin, "Power, influence and social circles: A new methodology for studying opinion makers," *American Sociological Review*, pp. 685–699, 1968.

[21] P. Zheng, Z. Huang, Y. Dou, and Y. Yan, "Rumor detection on social media through mining the social circles with high homogeneity," *Information Sciences*, vol. 642, p. 119083, 2023.

[22] Y. Yang, C. Fan, Y. Gong, W. Yeoh, and Y. Li, "Forwarding in Social Media: Forecasting Popularity of Public Opinion With Deep Learning," *IEEE Transactions on Computational Social Systems*, doi: 10.1109/TCSS.2024.3348018, 2024.

[23] C. S. Park and B. K. Kaye, "The tweet goes on: Interconnection of Twitter opinion leadership, network size, and civic engagement," *Computers in Human Behavior*, vol. 69, pp. 174–180, 2017.

[24] F. Nian and Z. Zhang, "The influence of opinion leaders on public opinion spread

and control strategies in online social networks," *IEEE Trans. Comput. Soc. Syst.*, Early Access, Mar. 2025. doi: https://doi.org/10.1109/TCSS.2025.3537701.

[25] X. Hao, D. Zheng, Q. Zeng, and W. Fan, "How to strengthen the social media interactivity of e-government: Evidence from China," *Online Information Review*, vol. 40, no. 1, pp. 79–96, 2016.

[26] M. M. Mostafa, "More than words: Social networks' text mining for consumer brand sentiments," *Expert Systems with Applications*, vol. 40, no. 10, pp. 4241–4251, 2013.

[27] Q. Kong, W. Mao, G. Chen, and D. Zeng, "Exploring trends and patterns of popularity stage evolution in social media," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 10, pp. 3817–3827, 2018.

[28] D. C. Mutz, "Impersonal influence: Effects of representations of public opinion on political attitudes," *Political Behavior*, vol. 14, pp. 89–122, 1992.

[29] Y. Hong, Y. Hu, and G. Burtch, "Embeddedness, Prosociality, and Social Influence," *MIS Quarterly*, vol. 42, no. 4, pp. 1211–A4, 2018.

[30] X. Fang and P. J. H. Hu, "Top persuader prediction for social networks," *MIS Quarterly*, vol. 42, no. 1, pp. 63–A14, 2018.

[31] Y. Lu, K. Jerath, and P. V. Singh, "The emergence of opinion leaders in a networked online community: A dyadic model with time dynamics and a heuristic for fast estimation," *Management Science*, vol. 59, no. 8, pp. 1783–1799, 2013.

[32] S. Deng, L. Huang, G. Xu, X. Wu, and Z. Wu, "On deep learning for trust-aware recommendations in social networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 5, pp. 1164–1177, May 2016.

[33] L. C. Freeman, "Turning a profit from mathematics: The case of social networks," *Journal of Mathematical Sociology*, vol. 10, nos. 3-4, pp. 343–360, 1984.

[34] M. Chandrashekar and F. C. Wong, "Thermodynamic systems analysis—I. A graph-theoretic approach," *Energy*, vol. 7, no. 6, pp. 539–566, 1982.

[35] M. Doostmohammadian and U. A. Khan, "Graph-theoretic distributed inference in social networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 4, pp. 613–623, 2014.

[36] P. Holme and J. Saramäki, "Temporal networks," *Physics Reports*, vol. 519, no. 3, pp. 97–125, 2012.

[37] M. Rodríguez-Ibánez, A. Casánez-Ventura, F. Castejón-Mateos, and P. M. Cuenca-Jiménez, "A review on sentiment analysis from social media platforms," *Expert Systems with Applications*, vol. 223, p. 119862, 2023.

[38] L. Ouyang, L. Yan, L. Liu, *et al.*, "Double-robust Bayesian variable selection and model prediction with spherically symmetric error," *IISE Transactions*, pp. 1–14, 2025.

[39] H. Kim, J. Tang, R. Anderson, and C. Mascolo, "Centrality prediction in dynamic human contact networks," *Computer Networks*, vol. 56, no. 3, pp. 983–996, 2012.

[40] C. C. Aggarwal and H. Wang, "Text mining in social networks," *Social Network Data Analytics*, pp. 353–378, 2011.

[41] W. Jia, R. Ma, L. Yan, W. Niu, and Z. Ma, "TT-graph: A new model for building social network graphs from texts with time series," *Expert Systems With Applications*, vol. 192, p. 116405, 2022.

[42] H. Wang, C. K. Chang, H. I. Yang, and Y. Chen, "Estimating the relative importance of nodes in social networks," *Journal of Information Processing*, vol. 21, no. 3, pp. 414–422, 2013.

[43] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electronic Markets*, vol. 31, no. 3, pp. 685–695, 2021.

[44] E. Elakkiya and S. Selvakumar, "Stratified hyperparameters optimization of feed-forward neural network for social network spam detection (SON2S)," *Soft Computing*, vol. 26, no. 21, pp. 11915–11934, 2022.

[45] M. Ramot and A. Martin, "Closed-loop neuromodulation for studying spontaneous activity and causality," *Trends in Cognitive Sciences*, vol. 26, no. 4, pp. 290–299, 2022.

[46] S. Bhagat, G. Cormode, and S. Muthukrishnan, "Node classification in social networks," *Social Network Data Analytics*, pp. 115–148, 2011.

[47] M. Aghaabbasi, M. Ali, M. Jasiński, Z. Leonowicz, and T. Novák, "On hyperparameter optimization of machine learning methods using a Bayesian optimization algorithm to predict work travel mode choice," *IEEE Access*, vol. 11, pp. 19762–19774, 2023.

[48] L. Ouyang, Y. Che, C. Park, et al., "A novel active learning Gaussian process modeling-based method for time-dependent reliability analysis considering mixed variables," *Reliability Engineering & System Safety*, vol. 244, p. 109916, 2024.

[49] L. Ouyang, Y. Che, L. Yan, et al., "Multiple perspectives on analyzing risk factors in FMEA," *Computers in Industry*, vol. 141, p. 103712, 2022.

[50] P. Xu, Y. Wu, E. Wei, T. Q. Peng, S. Liu, J. J. Zhu, and H. Qu, "Visual analysis of topic competition on social media," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2012–2021, 2013.

[51] C. Cheng, Z. Yu, Y. Zhang, and L. Feng, "Hybrid network using dynamic graph convolution and temporal self-attention for EEG-based emotion recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 12, pp. 18565–18575, Dec. 2024.

[52] K. Zhang, S. Bhattacharyya, and S. Ram, "Large-scale network analysis for online social brand advertising," *MIS Quarterly*, vol. 40, no. 4, pp. 849–868, 2016.