

Siwei LI, Jichao LI, Chang GONG, Tianyang LEI, Kewei YANG

Locating the source of diffusion in the early stage utilizing network monitors and graph convolutional networks

© Higher Education Press 2025

Abstract Locating the source of diffusion in complex networks is a critical and challenging problem, exemplified by tasks such as identifying the origin of power grid faults or detecting the source of computer viruses. The accuracy of source localization in most existing methods is highly dependent on the number of infected nodes. When there are few infected nodes in the network, the accuracy is relatively limited. This poses a major challenge in identifying the source in the early stages of diffusion. This article presents a novel deep learning-based model for source localization under limited information conditions, denoted as GCN-MSL (Graph Convolutional Networks and network Monitor-based Source Localization model). The GCN-MSL model is less affected by the number of infected nodes and enables the efficient identification of the diffusion source in the early stages. First, pre-deployed monitor nodes, controlled by the network administrator, continuously report real-time data, including node states and the arrival time of anomalous signals. These data, along with the network topology, are used to construct node features. Graph convolutional networks are employed to aggregate information from multiple-order neighbors, thereby forming comprehensive node representations. Subsequently, the model is trained with the true source labeled as the target, allowing it to distinguish the source node from other nodes within the network. Once trained, the model can be applied to locate hidden sources in other diffusion networks. Experimental results across multiple data sets demonstrate the superiority of the GCN-MSL model, especially in the early stages of diffusion, where it significantly enhances both the accuracy and efficiency of

source localization. Additionally, the GCN-MSL model exhibits strong robustness and adaptability to variations in external parameters of monitor nodes. The proposed method holds significant value in the timely detection of anomalous signals within complex networks and preventing the spread of harmful information.

Keywords source localization, graph convolutional network, network monitor, information diffusion

1 Introduction

Information diffusion is a prevalent phenomenon in complex networks, occurring across various systems such as industrial systems, smart grids, social media platforms, and communication networks. As information spreads through these networks, it can yield both beneficial outcomes and harmful consequences. Accurate and efficient identification of the source of harmful information is critical, as it enables timely intervention, minimizes negative impacts, and ensures the normal operation of the network. This is particularly important when addressing rapidly evolving threats, where delays in detection and response can exacerbate the consequences. In the field of smart manufacturing, the growing demand for small-batch and highly customized products has led to increasingly complex and interconnected factory production lines, integrated with various embedded processors, forming smart manufacturing networks (Guo et al., 2023). A failure in any part of the network can adversely impact related processes and may even result in the collapse of the entire manufacturing network. Quickly identifying the source of the failure in the early stages and promptly adjusting production strategies are essential for maintaining the stable operation of the manufacturing network. The public accessibility and inadequate security of devices render the smart grid susceptible to a variety of malicious threats and cyberattacks. Identifying the source of smart grid failures (Li and Jia, 2021) as soon as possible

Received Nov. 27, 2024; revised Jan. 26, 2025; accepted Mar. 2, 2025

Siwei LI, Jichao LI (✉), Chang GONG, Tianyang LEI, Kewei YANG
College of Systems Engineering, National University of Defense Technology, Changsha 410073, China
E-mail: ljcnudt@hotmail.com

This research was supported in part by the National Natural Science Foundation of China (Grant Nos. 72371244, 72231011, 72301286, 72431011 and 72421002).

helps minimize losses and damage, making it a critical issue that impacts the national economy and public livelihoods. The rapid development of social software has led to the expansion of social networks and accelerated the dissemination of information. False information spreads rapidly through these networks, resulting in detrimental effects (Meel and Vishwakarma, 2020). Detecting the source of rumors in social networks in the early stages of diffusion can help develop effective strategies to sever critical paths of rumor propagation and control their spread promptly. Furthermore, identifying Patient Zero in an epidemic (Zhao and Cheong, 2023) and pinpointing the source of computer viruses (Tang et al., 2024) as early as possible are critical issues closely related to public health, property, and information security. These issues all fall within the domain of source localization in networks.

Numerous scholars have conducted extensive research on the source localization problem and proposed various methods grounded in network science. Source localization methods can be classified into three categories based on the type of observed information: complete observation, snapshot observation, and monitor observation (Shelke and Attar, 2019). Complete observation provides state information for all nodes in the network, whereas snapshot observation only provides the state of a subset of nodes. Monitor-based methods involve pre-selecting certain nodes as monitors in the network to record observed information, such as node states and the time of receiving signals. These methods then perform source inference based on this information and network structure, utilizing techniques such as Pearson correlation centrality (Wang and Sun, 2020) and Gaussian estimators (Yang et al., 2020). Paluch et al. (2020) demonstrated that the most effective source localization methods rely on monitor observation.

Early detection of the source helps minimize the impact of harmful information propagation. However, during the early stages of diffusion, the number of monitor nodes that have received information is typically limited. Most existing methods primarily rely on monitor nodes that have received signals for source inference, and the limited available information significantly reduces the accuracy of source localization. To accurately locate the diffusion source with minimal delay and seize the optimal opportunity to control diffusion, this paper proposes a method based on Graph Convolutional Networks (GCNs). The method leverages all monitors to help infer diffusion sources, including monitor nodes that have not received signals as well as those that have, thereby improving the accuracy during the early stage. The main contributions of this work are summarized in three key aspects.

(1) Develop a novel source localization framework based on GCN, referred to as GCN-MSL. This framework treats the source localization problem based on monitor

observation as a classification task, leveraging the robust learning capabilities of neural networks to differentiate the source node from other nodes in the network.

(2) Propose a node feature extraction method that fully exploits information collected from all monitor nodes and network topology details. This approach ensures that the accuracy of source localization remains unaffected by the spread time. The ablation study confirms the effectiveness of the selected node features.

(3) Conduct experiments on both real and synthetic data sets. Experimental results demonstrate that the proposed method achieves high accuracy in source localization, performing particularly well during the early stages of diffusion. The tests evaluating the model's generalization ability demonstrate its robustness in locating source nodes.

The following sections of this paper are organized as follows: Section 2 reviews related work in source localization. Section 3 formulates the source localization problem. Section 4 introduces the proposed methodology, including the overall framework and technical details. In Section 5, extensive experiments are conducted to evaluate the accuracy and generalization capability of the model. Finally, Section 6 summarizes our work and discusses potential future research directions.

2 Related work

This section provides a literature review on the source localization problem, summarizing existing research in the field from two perspectives: classical methods and Graph Neural Networks (GNNs)-based methods.

2.1 Classic method

As mentioned earlier, source localization methods can be classified into three categories based on the type of network observation: complete observation, snapshot observation, and monitor observation.

Complete observation provides the state information of all nodes in the network at a given moment. Shah and Zaman (2011) proposed a method for source identification in tree-structured networks, where they constructed a rumor centrality metric to estimate the potential source of the information spread. LPSI (Wang et al., 2017) is the first method designed to perform multi-source localization without knowledge of the underlying propagation model; it leverages the concept of source prominence and estimates the source through iterative label propagation among nodes.

However, obtaining accurate state information for all nodes is often a challenging task. Snapshot observation provides detailed data for specific infected nodes at particular moments. By aggregating multiple snapshots over different time periods, sufficient information for

source localization can be accumulated (Cai et al., 2018). Zhou et al. (2021) proposed a two-stage algorithm using semi-supervised learning and label ranking to address the source localization problem with snapshots. Jiang et al. (2024b) introduced an improved Dynamic Message Passing algorithm (S-DMP) for source localization in signed networks.

Deploying monitor nodes within the network is an effective approach for observation. Using the data collected from monitor nodes for source inference can reduce the cost of acquiring global network state information. Pinto et al. (2012) developed a maximum likelihood estimator, suggesting that the node minimizing the difference between the monitor observation delay and the estimated delay is the most likely source of propagation. Paluch et al. (2018) enhanced this algorithm by excluding monitors with low-quality information. Zhu and Ying (2016) presented a path-based source identification algorithm. By utilizing the obtained snapshots and the infection paths recorded by monitors, an iterative process is conducted to identify the nodes with local maximum labels. Wang et al. (2023) designed a greedy coverage-based method, enabling rapid source localization. Pan et al. (2024) proposed an innovative Hill-Climbing algorithm that reduces computational time while maintaining high accuracy in source localization.

2.2 GNN-based method

GNN is a deep learning-based method that operates in the graph domain, capturing the dependencies of graph structures through message passing between nodes (Zhou et al., 2019). Based on the convolution operation, GNN are broadly categorized into spectral methods and spatial methods. Spectral methods include Chebyshev Spectral CNN (ChebNet) (Michaël et al., 2016) and Graph Convolutional Networks (GCN) (Kipf and Welling, 2017), etc. Spatial methods include GraphSAGE (William et al., 2017) and Backtrackless Aligned-Spatial Graph Convolutional Networks (BASGCN) (Bai et al., 2022), etc.

Due to its powerful ability to handle graph-structured data, GNN has increasingly been employed as a key method for source localization in complex networks. Dong et al. (2019) utilized GCN to enhance the performance of LPSI. Wang et al. (2022) proposed the Invertible Validity-aware Graph Diffusion model (IVGD), which leverages graph residual networks to infer diffusion sources. Ling et al. (2022) introduced the Source Localization Variational AutoEncoder (SL-VAE), a deep generative model that approximates the distribution of diffusion sources, enabling source localization under various diffusion patterns. Dong et al. (2022) proposed a sequence-to-sequence model, Graph Constraint-based Sequential Source Identification (GCSSI), which uses encoder-decoder architectures and graph constraint-based multi-task learning to predict sources in an end-to-end

manner. Xu et al. (2024) designed a probabilistic framework incorporating continuous normalizing flows with invertible transformations and GNN to explicitly model the uncertainty of diffusion sources. Cheng et al. (2024) proposed a heuristic framework for sources detection in social networks via GCN.

The above provides a brief overview of the field of source localization. The strong learning capabilities of GNN have significantly improved the accuracy of source localization under both complete and snapshot observation conditions. However, to the best of our knowledge, existing monitor-based approaches predominantly rely on classic mathematical frameworks. Most of these methods necessitate waiting until monitor nodes in the network accumulate sufficient information to enable accurate source localization. Few methods can effectively identify the source using the limited information available from monitor nodes in the early stages of diffusion. Furthermore, methods relying on complete or snapshot observation also encounter difficulties in achieving high localization accuracy in the early stages. Hence, there is a critical need to develop methods capable of rapidly localizing the source in the early stages of diffusion. Source localization can be framed as a classification task, where nodes in the graph are classified as either source nodes or others. GCN (Kipf and Welling, 2017), a widely applied type of graph neural network, effectively captures neighborhood information of nodes. It is characterized by a relatively low number of model parameters and quick training speeds, making it well-suited for tasks such as node and graph classification. Investigating how GCN can be adapted to monitor-based scenarios, in order to better leverage data from monitor nodes and accurately infer the hidden source in the early stages of diffusion, presents an intriguing and valuable area of research.

3 Problem formulation

The source localization problem is the inverse problem of maximizing the influence of nodes in the propagation process (Ou et al., 2022), aiming to identify the original node from which the diffusion originated. It is assumed that any node in the network is capable of deploying a monitor, with no technical constraints. Additionally, it is assumed that information propagates across the network following a fixed pattern, without the fusion of multiple propagation models. Based on these assumptions, we present the formulation of the source localization problem.

First, in the context of monitor-based source localization, it is crucial to implement a strategy, such as centrality-based or random selection methods, to deploy a sufficient number of monitors in the network. A node is then randomly selected as the information diffusion source, and a specific propagation model is used to simulate

the spread of information across the network. During the propagation process, monitor nodes collect information, including their own states and activation times. Based on the collected information, various algorithms are applied to locate the source. At this stage, specific accuracy metrics can be used to evaluate the performance of the algorithms. This problem can be formally expressed using the following formulas. Table 1 presents the symbols used in the formulas along with their definitions. Figure 1 illustrates a schematic representation of a network equipped with monitors.

Given a network $G = (V, E)$, where V is the set of nodes and E is the set of edges. The set of monitors is denoted as M , which is a subset of V . ξ indicates the proportion of monitors in all nodes ($\xi \in [0, 1]$, $M = \{m_i\}_{i=1}^{|V| \times \xi}$). Assuming that node s is the actual infection source, after the propagation ends, the set of information received by the monitors is inf . Then the estimated source \hat{s} is the node maximizes the probability of observations inf , which can be expressed as:

$$\hat{s} = \operatorname{argmax}_{v_i \in V} P(inf | s = v_i). \quad (1)$$

Table 1 Symbols and definitions

Symbol	Definition
G	The given network
V	The set of nodes
E	The set of edges
M	The set of monitors
ξ	The proportion of monitors in all nodes
s	The actual source
inf	The set of information received by monitors
\hat{s}	The estimated source

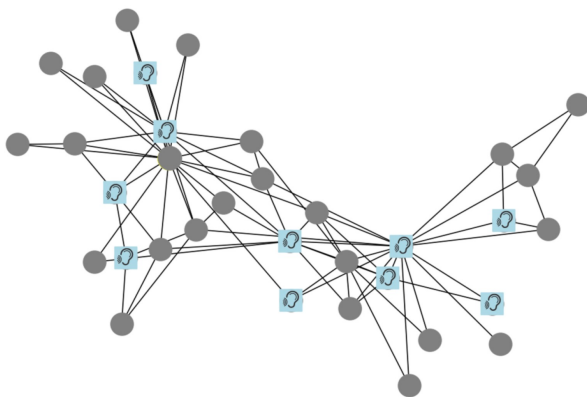


Fig. 1 A schematic illustration of a network equipped with monitors. The blue squares with ear-shaped patterns in the figure represent nodes equipped with monitors, whereas the gray circles represent regular nodes.

4 GCN and network monitor-based source localization model

This section presents the overall framework for the GCNs and network monitor-based source localization model (GCN-MSL), as illustrated in Fig. 2. The framework comprises three primary modules: Data Generation, Feature Extraction, and Model Training and Optimization. The Data Generation module serves as a preparatory step preceding the execution of the algorithm, while Feature Extraction, Model Training and Optimization constitute the core components of the GCN-MSL algorithm. A detailed explanation of each module is provided below.

4.1 Data generation

During the data generation stage, an initial network G is selected, and a specific number of monitor nodes are deployed within it. The placement of monitor nodes can be random or follow specific strategies, such as the centrality strategy. While centrality-based strategies may improve localization accuracy by increasing the amount of information collected by the monitors, the placement of monitor nodes in real-world scenarios is not necessarily uniform. These strategies may be based on betweenness centrality, degree centrality, or even random placement. To ensure the model's adaptability to various monitor placement strategies, this study employs a random placement strategy to generate the training data sets.

One node is randomly selected as the source node s , and the message diffusion process from the source node through the network in a snowballing manner is simulated. In this context, "snowballing" refers to the process whereby, once a node is activated, the message is passed along the edges to all its neighboring nodes, with the transmission time on each edge following a Gaussian distribution. The propagation cut-off time t_p corresponds to the moment when the anomaly is detected by the administrator. Monitor nodes report their state at time t_p (whether they have received the information) and the time at which they receive the message. Nodes that have received the information are referred to as activated monitor nodes, while nodes that have not received the information are referred to as inactivated monitor nodes.

A series of simulated diffusion processes are conducted on the network G to generate a collection of infection network samples, which are subsequently divided into training, validation, and test sets.

4.2 Feature extraction

This study selects four features for each node based on the information that monitors can collect and the network topology. First, the information that monitor nodes can collect includes both their own state and the time at

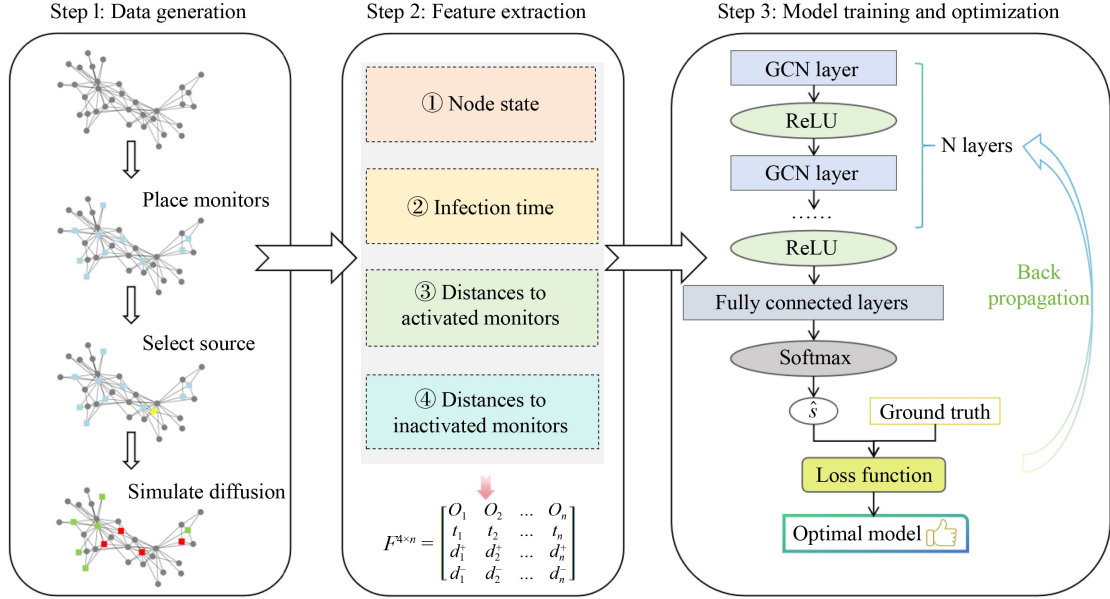


Fig. 2 The overall framework of GCN-MSL. In the Data Generation part, the blue squares represent monitor nodes, the red squares denote monitor nodes that have received information, the green squares indicate monitor nodes that have not received information, the yellow circle signifies the source node and the gray circles represent nodes with unknown states.

which the information is received. Therefore, the node state and the time of information reception are selected as the first two features for each node. Second, the source node should naturally be closer to activated monitor nodes and farther from inactivated ones. Hence, the sum of distances from each node to all activated monitor nodes, as well as the sum of distances to inactivated monitor nodes, are selected as the third and fourth features, respectively. These two features utilize the network topology information. A detailed explanation is provided below.

(1) Node state O_i : The states of activated monitor nodes (i.e., those that have received information) are assigned a value of 1; the states of inactivated monitor nodes (i.e., those that have not received information) are assigned a value of -1 ; the states of other nodes are considered unknown, represented by a value of 0.

(2) Infection time t_i : The infection time for activated monitor nodes corresponds to the actual time at which they receive information. For inactivated monitor nodes and other nodes with unknown states, the infection time is assigned a value of 0.

(3) Distances to activated monitors d_i^+ : This metric represents the sum of the shortest path lengths from a given node to all activated monitor nodes. Intuitively, the source node is expected to be closer to the nodes that have received information and farther from those that have not. The calculation is given by:

$$d_i^+ = \sum_{v_j \in M^+} d_{ij}, \quad (2)$$

where M^+ is the set of activated monitor nodes and d_{ij} is

the shortest path length between nodes v_i and v_j .

(4) Distances to inactivated monitors d_i^- : This metric represents the sum of the shortest path lengths from each node v_i to all inactivated monitor nodes. The calculation is given by:

$$d_i^- = \sum_{v_j \in M^-} d_{ij}, \quad (3)$$

where M^- is the set of inactivated monitor nodes and d_{ij} is the shortest path length between nodes v_i and v_j .

By aggregating the four-dimensional features of all nodes, we form a $|V| \times 4$ matrix as the input for the model training and optimization module, where $|V|$ represents the total number of nodes in the network.

4.3 Model training and optimization

The model training and optimization module is designed using GCN layers and fully connected layers, with the results being output through a Softmax classifier.

GCN extends the convolution operation from images to graph-structured data by aggregating the features of a node v_i and its neighbors to learn a function that generates a representation for the node v_i (Wu et al., 2021). GCN can be applied to a variety of tasks, including node classification, graph classification, and link prediction.

The inter-layer propagation rule of multi-layer GCN is as follows:

$$H^{(l+1)} = \sigma(\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}), \quad (4)$$

$$\hat{A} = A + I, \quad (5)$$

where I is the identity matrix, A is the adjacency matrix of graph G , \hat{A} represents the adjacency matrix of graph G with self-loops, $\hat{D}_{ii} = \sum_j \hat{A}_{ij}$ is the Laplacian matrix of G , $H^{(l)}$ is the feature matrix from layer l , $H^{(0)} = X$ is the initially generated input feature, $W^{(l)}$ is the trainable parameter matrix for the convolution operation at layer l , and $\sigma(\cdot)$ is the activation function.

Each convolution operation captures the features of neighboring nodes at an additional layer (Peng et al., 2022). We employ multiple GCN layers to aggregate features from multi-hop neighboring nodes, using ReLU as the activation function. After passing through the fully connected layers (FC layers), the output matrix from the GCN layers is converted into a vector of length equal to the number of nodes. The Softmax classifier then outputs the probability of each node being the source, with the node having the highest probability serving as the predicted source \hat{s} , which also corresponds to the label of the infection graph. We compute the average loss between the true labels of the samples and the outputs to update the weight matrix, saving the model that performs best on the validation set as the optimal model.

In this study, the source identification problem is framed as a classification task. We adopt cross-entropy loss as the loss function and apply L2 regularization to mitigate overfitting. The loss function is calculated as follows:

$$Loss(\hat{l}, l) = - \sum_i^{|V|} l_i \log(\hat{l}_i) + \lambda \|\omega\|_2, \quad (6)$$

where $|V|$ is the number of nodes in the graph; $l = (l_1, l_2, \dots, l_{|V|})$ is the ground truth vector with a single nonzero element which is 1, corresponding to the true source node of propagation, while the remaining elements are all 0; \hat{l} is the estimated vector given by the trained model. Additionally, ω denotes the parameter vector composed of all trainable parameters in GCN, and $\|\omega\|_2$ represents the L2 regularization term of this parameter vector with a weight coefficient as λ .

5 Experiments and analysis

This section presents the performance of GCN-MSL on both real-world and synthetic networks. We conduct comparative experiments with several baseline methods and assess the model's generalization ability concerning monitor density and deployment patterns. Additionally, we design ablation experiments to investigate the impact of the four node features.

5.1 Data sets

As shown in Table 2, four data sets are selected to evaluate the performance of our algorithm: two real-world data

sets and two synthetic data sets. Figure 3 presents the network structures of these four data sets.

(1) **Karate** (Wang et al., 2022) is a social network representing friendships among 34 members of a karate club at a university in the United States.

(2) **Dolphin** (Wang et al., 2022) is a social network of 62 dolphins in Doubtful Sound, New Zealand.

(3) **BA500** is a simulated scale-free network with 500 nodes generated using the Barabási-Albert model, where each new node connects to three existing nodes.

(4) **WS1000** is a simulated small-world network with 1000 nodes generated using the Watts-Strogatz model, where each node initially has five neighbors and a random reconnection probability of 0.2.

5.2 Baselines

To evaluate the effectiveness of GCN-MSL, it is compared with several classic methods as follows:

(1) **LPTV** (Pinto et al., 2012; Gong et al., 2024): A method for source localization that relies on the difference between the “observed delay” and “estimated delay” of monitor nodes, assuming that the propagation time on each edge follows a Gaussian distribution.

(2) **TRBS** (Shen et al., 2016; Jin et al., 2023): A time-reversal backward spreading algorithm that identifies topologically shortest paths from monitor nodes to potential sources and compares them with observed reception time.

Table 2 Overview of the data sets

Data set	Number of nodes	Number of edges	Average degree
Karate	34	78	4.58
Dolphin	62	159	5.13
BA500	500	1491	5.96
WS1000	1000	2000	4.00

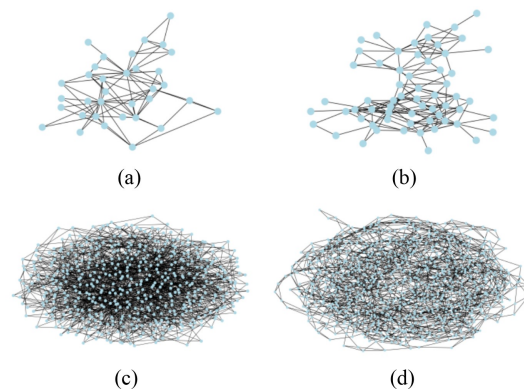


Fig. 3 Network structure of each data set. (a) Karate: a social network with 34 nodes; (b) Dolphin: a social network with 62 nodes; (c) BA500: a scale-free network with 500 nodes; (d) WS1000: a small-world network with 1000 nodes.

(3) **GMLA** (Paluch et al., 2018; Jiang et al., 2024a): An improved version of LPTV that discards monitors with low-quality information (i.e., those with large spread encounter times). Potential sources are selected based on the likelihood gradient derived from high-quality monitors.

(4) **Jordan Center** (Zhu and Ying, 2016; Hou et al., 2024): This algorithm identifies the node with the minimum infection eccentricity as the infection center. A node’s infection eccentricity is defined as the maximum shortest path length from the node to all infected nodes.

5.3 Experiment settings

This section describes the experimental methods, algorithm parameter settings, and evaluation metrics.

In the data generation stage, a node is randomly selected as the propagation source in the network. Next, a subset of nodes in the network is randomly selected to serve as monitor nodes. Having an excessive number of monitor nodes in the network is unrealistic, whereas too few would fail to provide sufficient information. Referring to the proportion of monitor nodes used in previous studies, we selected 30% of the nodes as monitor nodes. The message spreads through the network in a snowballing manner, meaning that when a node is activated, the message propagates along the edges to all its neighboring nodes. The propagation delay on each edge is represented by a random variable $\theta \sim N(1, 0.5^2)$. To ensure the nonnegative value of the time delay, set $\theta > 0$. The propagation time range is set according to the network size. The minimum propagation time t_{\min} is uniformly set to 1.2, while the maximum propagation time t_{\max} is determined by the time it takes for most nodes in the network to receive the message. A propagation cut-off time is randomly selected between t_{\min} and t_{\max} . The above steps are repeated multiple times on the same network to generate a series of samples, with 80% used as training samples and 20% as validation samples.

GCN-MSL is a deep learning-based model. To mitigate overfitting, dropout is applied after the hidden layers, and the Adam optimizer (Kingma and Ba, 2015; Guo et al., 2021) is used to optimize the model parameters. The specific parameter settings are presented in Table 3.

The generation method for the test data are similar to that of the training data. To evaluate the performance of the proposed algorithm throughout the propagation process, we fix the propagation cut-off time and generate 1,000 samples for each test data set. Based on the spread time range of different data sets, multiple sets of test data are generated to observe the impact of spread time on algorithm performance.

For GCN-MSL, the probability vector outputted by the trained model is sorted to obtain the sequence of possible sources. For the baseline methods, the output is a sequence of potential sources, ranked in descending order

Table 3 Default settings of important hyper-parameters

Data set	Learning rate	GCN layers	FC layers	Hidden size	Dropout rate	Epoch
Karate	0.001	2	2	512	0.001	100
Dolphin	0.001	2	2	512	0.001	100
BA500	0.001	2	2	512	0.001	100
WS1000	0.0001	2	2	512	0.001	200

of likelihood, with the highest-ranked potential source being the predicted source.

We use the following three metrics to evaluate the performance of the proposed model:

(1) **Accuracy**: The proportion of test samples in which the true diffusion source is correctly identified out of all test samples. This is a positive indicator.

(2) **$\gamma\%$ -accuracy**: The proportion of test samples where the actual source appears within the top $\gamma\%$ of the predicted sequence. This is a positive indicator. Since exact identification of the source is not always required, providing a range of potential diffusion sources with a certain tolerance level is also meaningful. In this study, different values of γ are set for each data set according to the network size.

(3) **Rank of the actual source (Average Rank)**: The average rank of the actual source within the predicted source sequence. This is a negative indicator, and a smaller value indicates better performance of the method.

5.4 Results and analysis

This study evaluates the source localization performance of various methods on synthetic and real network data sets, comparing our approach with other baseline methods. The performance of the various methods is shown in Tables 4–7 and Fig. 4. Tables 4–7 list the results for only the first three time points, representing the early stage of diffusion. Larger values of Accuracy and $\gamma\%$ -accuracy indicate better performance, while a smaller Average Rank suggests superior performance. In Fig. 4, the first column contains four images showing the Accuracy curves of different algorithms over time across four data sets. The second column displays the $\gamma\%$ -accuracy curves over time. The third column presents the Average Rank curves, with lower values are preferable.

As observed, GCN-MSL significantly outperforms other algorithms during the early stage of diffusion, and its performance remains relatively stable throughout the entire diffusion process, unaffected by the spread time. Furthermore, our proposed algorithm demonstrates robust performance across networks of varying sizes and types, suggesting its broad applicability. In the Karate data set, GCN-MSL consistently exhibits stable performance. Although LPTV surpasses GCN-MSL in the later stages of diffusion in terms of $\gamma\%$ -accuracy and Average Rank, GCN-MSL maintains significant advantages in the early

Table 4 Results in the early stage (Karate)

Method	Accuracy ^{a)}			$\gamma\%$ -accuracy ^{b)}			Average Rank ^{c)}		
	T1	T2	T3	T1	T2	T3	T1	T2	T3
GCN-MSL	0.219	0.234	0.232	0.408	0.405	0.411	7.510	7.393	7.286
LPTV	0.115	0.137	0.157	0.263	0.318	0.308	11.320	10.008	9.970
TRBS	0.119	0.143	0.154	0.239	0.310	0.307	11.417	9.995	9.837
GMLA	0.119	0.127	0.141	0.272	0.293	0.308	11.111	10.303	10.112
Jordan Center	0.120	0.097	0.081	0.295	0.281	0.255	9.191	9.169	9.610

Note: ^{a)} Positive indicator; ^{b)} positive indicator, $\gamma = 10$; ^{c)} negative indicator

Table 5 Results in the early stage (Dolphin)

Method	Accuracy ^{a)}			$\gamma\%$ -accuracy ^{b)}			Average Rank ^{c)}		
	T1	T2	T3	T1	T2	T3	T1	T2	T3
GCN-MSL	0.243	0.262	0.316	0.645	0.683	0.719	6.802	6.003	5.362
LPTV	0.098	0.166	0.208	0.307	0.444	0.593	18.785	13.668	9.579
TRBS	0.093	0.144	0.196	0.306	0.430	0.581	19.267	14.142	10.268
GMLA	0.130	0.162	0.175	0.493	0.538	0.557	15.453	15.511	14.977
Jordan Center	0.147	0.119	0.088	0.515	0.454	0.386	9.407	10.584	12.263

Note: ^{a)} Positive indicator; ^{b)} positive indicator, $\gamma = 10$; ^{c)} negative indicator

Table 6 Results in the early stage (BA500)

Method	Accuracy ^{a)}			$\gamma\%$ -accuracy ^{b)}			Average Rank ^{c)}		
	T1	T2	T3	T1	T2	T3	T1	T2	T3
GCN-MSL	0.203	0.203	0.217	0.410	0.429	0.448	45.939	37.805	34.850
LPTV	0.007	0.012	0.019	0.020	0.043	0.057	199.829	170.202	144.818
TRBS	0.012	0.015	0.017	0.029	0.037	0.049	220.741	199.603	175.276
GMLA	0.064	0.051	0.050	0.130	0.135	0.143	108.459	119.588	114.593
Jordan Center	0.081	0.051	0.035	0.281	0.215	0.125	60.783	76.134	90.366

Note: ^{a)} Positive indicator; ^{b)} positive indicator, $\gamma = 1$; ^{c)} negative indicator

Table 7 Results in the early stage (WS1000)

Method	Accuracy ^{a)}			$\gamma\%$ -accuracy ^{b)}			Average Rank ^{c)}		
	T1	T2	T3	T1	T2	T3	T1	T2	T3
GCN-MSL	0.318	0.369	0.393	0.819	0.847	0.877	3.792	3.308	3.011
LPTV	0.021	0.048	0.087	0.038	0.088	0.132	305.888	223.175	177.155
TRBS	0.040	0.050	0.090	0.080	0.096	0.135	322.451	253.214	221.489
GMLA	0.252	0.305	0.322	0.708	0.750	0.787	96.202	88.952	67.921
Jordan Center	0.284	0.294	0.299	0.761	0.790	0.778	4.909	4.995	4.584

Note: ^{a)} Positive indicator; ^{b)} positive indicator, $\gamma = 0.5$; ^{c)} negative indicator

stage across all three evaluation metrics, with its accuracy more than 1.8 times higher than that of other algorithms. Results from the Dolphin data set show that GCN-MSL outperforms other algorithms during the early stage. As the diffusion progresses, the performance of LPTV, TRBS, and GMLA gradually improves. In the later stages of propagation, LPTV performs nearly on par with

GCN-MSL, while the performance of other algorithms continues to lag behind GCN-MSL. In the scale-free network with 500 nodes, our method consistently maintains a clear advantage across all three metrics. It excels particularly in accuracy, significantly outperforming other algorithms. In the small-world network, our method also demonstrates its strengths. Experimental results indicate

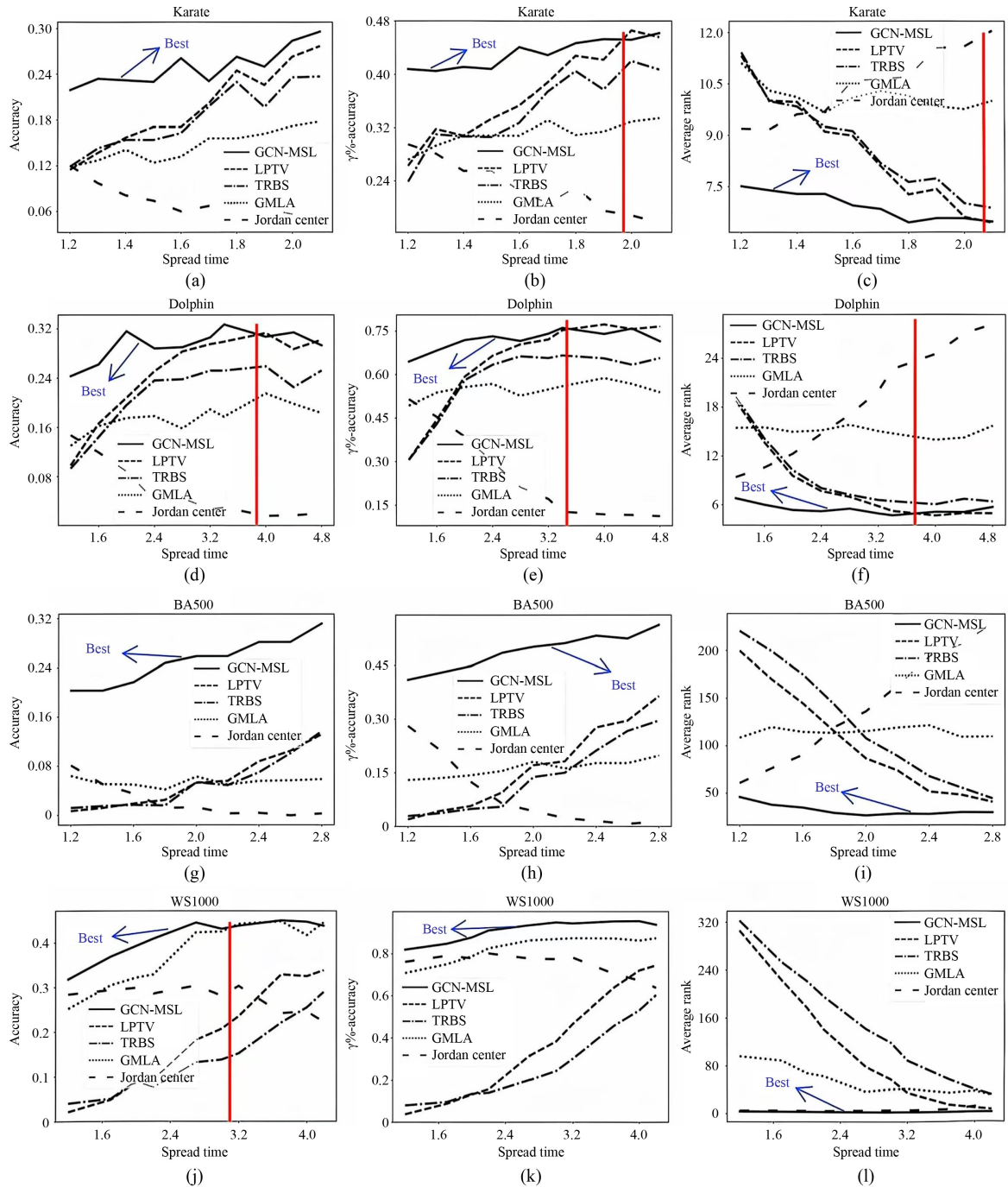


Fig. 4 Experiment results. (a)–(c): Results in Karate; (d)–(f): results in Dolphin; (g)–(i): results in BA500; (j)–(l): results in WS1000. In each subfigure, the area to the left of the red line indicates the region where GCN-MSL outperforms the other methods. If no red line is present, it indicates that GCN-MSL consistently maintains an advantage.

that GCN-MSL exhibits exceptional performance in $\gamma\%$ -accuracy and Average Rank. Throughout the entire diffusion process, $\gamma\%$ -accuracy consistently remains above 0.8, and the Average Rank stays below 4, effectively enabling us to narrow down the potential source early in the process. While GMLA is well-suited for small-world networks and surpasses our model in accuracy, our approach retains its superiority in $\gamma\%$ -accuracy and

Average Rank.

GCN-MSL maintains stable performance throughout the diffusion process, while the performance of most baseline methods is significantly influenced by the spread time. This is because the inference of these methods depends on the information provided by activated monitors. However, in the early stage of diffusion, only a few monitor nodes have received information. As a result,

there is limited information available, leading to low source localization accuracy. In contrast, our model can fully utilize all monitor nodes, including those that have already received information and those that have not, by leveraging neural networks to learn the relationships between the information provided by the monitors, the network topology, and the source node. Consequently, it achieves high source localization accuracy, even during the early stage of diffusion. We observe that as the network size increases, the advantages of GCN-MSL become more pronounced. This may be due to the propagation delay of information on edges being a random variable that follows a Gaussian distribution. As the network size increases, the uncertainty in accumulated propagation delays grows, leading to a decline in localization accuracy for classical methods. In contrast, GCN-MSL is less affected by this randomness, as GCN enables us to learn the pattern of information propagation from the data sets.

Unfortunately, GCN-MSL, being a deep learning-based method, requires time for parameter selection and training. However, once training is complete, using the model for source localization takes minimal time. For instance, in the WS1000 network, when predicting on the test data set with a diffusion cut-off time of 3.7, GCN-MSL takes an average of 0.0067 s, while LPTV takes 0.0656 s, TRBS takes 0.0075 s, GMLA takes 0.0080 s, and Jordan Center takes 0.0101 s. Additionally, our experiments are conducted using a specific information propagation model, without considering other models such as the Independent Cascade (IC) model, Linear Threshold (LT) model, or SIR disease propagation model. As a result, the conclusions drawn from the experiments are subject to certain limitations. Furthermore, the networks in the experiments have a fixed structure. In real-world applications, dynamic networks may exist, which would render our model unsuitable.

5.5 Generalization ability study

For each network, we vary the deployment density of monitor nodes to evaluate the model's generalization ability. The monitor density is set to 25%, 28%, 33%, and 35%, and new test data are generated for each density scenario. The previously trained models are then applied to identify the diffusion sources, with results presented in Fig. 5. The red curve represents the baseline, which corresponds to a monitor density of 30%.

It can be observed that reducing the number of monitor nodes slightly decreases the model's prediction accuracy, while increasing the number of monitor nodes leads to improved accuracy. Overall, the model's performance remains stable, indicating that it is relatively unaffected by moderate changes in the density of monitor nodes. This suggests that the previously trained model can accommodate small variations in monitor node density.

Next, we change the deployment strategy of monitor nodes in the network to further test our model. Monitor nodes are deployed based on the betweenness centrality of nodes in descending order, with the number of monitor nodes set to 30% of the total nodes. The test results are shown in Fig. 6.

Compared to the results in Fig. 4, GCN-MSL exhibits a slight decrease in performance, particularly on smaller data sets, due to its training being based on the random deployment of monitors. Nevertheless, the proposed model continues to deliver impressive performance, maintaining a clear advantage in the early stages of diffusion. In the later stages, it is surpassed by individual methods on certain data sets. Overall, GCN-MSL performs consistently well across all data sets.

5.6 Ablation study

The framework proposed in this paper selects four features for each node: (1) node state O_i , (2) infection time t_i , (3) sum of distances to all activated monitors d_i^+ , and (4) sum of distances to all inactivated monitors d_i^- . To validate the effectiveness of these selected features, we conduct an ablation study on four data sets. Eight ablation models are designed for the purpose of this study. The first ablation model is trained without feature (4), using only features (1), (2), and (3), referred to as *w/o d⁻*. The second ablation model is trained without feature (3), called *w/o d⁺*. The third model omits feature (2), referred to as *w/o t*. The fourth model omits feature (1), denoted as *w/o O*. Additional models include model *only O*, which uses only feature (1); model *only t*, which uses only feature (2); model *only d⁺*, which uses only feature (3); and model *only d⁻*, which utilizes only feature (4). The original model, which includes all four features, is denoted as *Orig*.

Figure 7 presents the results of the ablation study conducted across four data sets. The results indicate that the original model outperforms all other models across all data sets, followed by *w/o d⁻* and *w/o d⁺*. Next is *w/o O*, which demonstrates strong performance in WS1000. Model *w/o t* exhibits better performance in the Karate data set. The performance of the remaining models is inconsistent and often unstable, with some models struggling to learn effectively. In these cases, the output probability values for all nodes may either be identical or randomly distributed, making it difficult to extract meaningful patterns. This issue is particularly evident in model *only d⁻*. These findings suggest that the four selected features are essential for enabling the model to distinguish between the source node and others, with each feature playing a critical role. Among the four Features, (1) node state and (2) infection time are the most significant. Compared to the ablation models, the original model makes better use of the information provided by monitor nodes, resulting in superior source localization perfor-

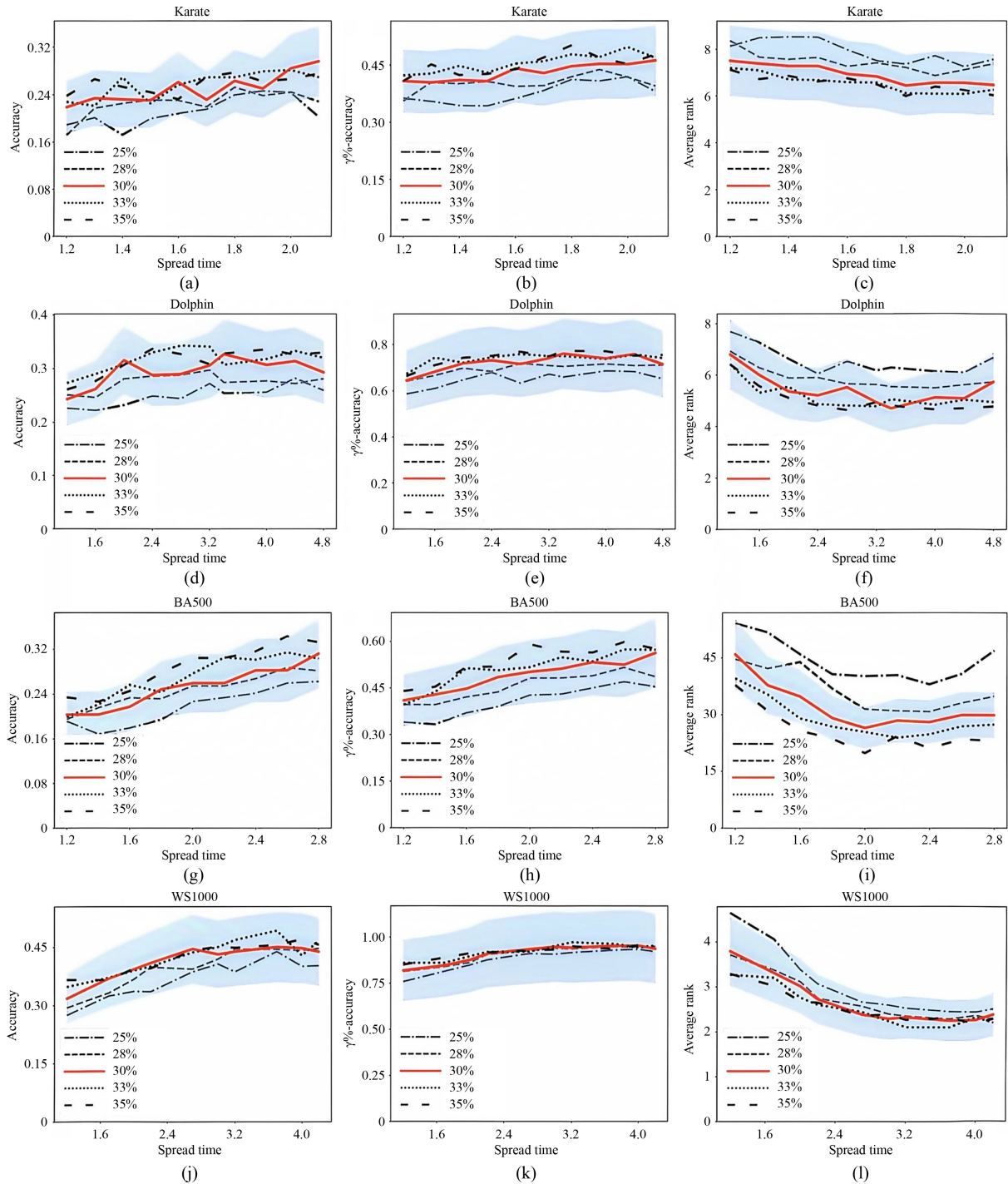


Fig. 5 Results of altering the monitor density. (a)–(c): Results in Karate; (d)–(f): results in Dolphin; (g)–(i): results in BA500; (j)–(l): results in WS1000. The red curve corresponds to a monitor node density of 30%, which is the density used in the training set. The blue shading represents a variation range of $\pm 20\%$ around the red curve.

mance. This is also one of the key reasons why our approach outperforms the baseline methods.

6 Conclusions

This paper introduces GCN to the monitor-based source

localization problem, presenting the GCN-MSL model. We select four features to construct the input feature matrix of the model and aggregate the features of neighboring nodes through multiple GCN layers. Experimental results demonstrate that the GCN-MSL method significantly enhances source localization accuracy, enabling the rapid identification of diffusion sources in the early

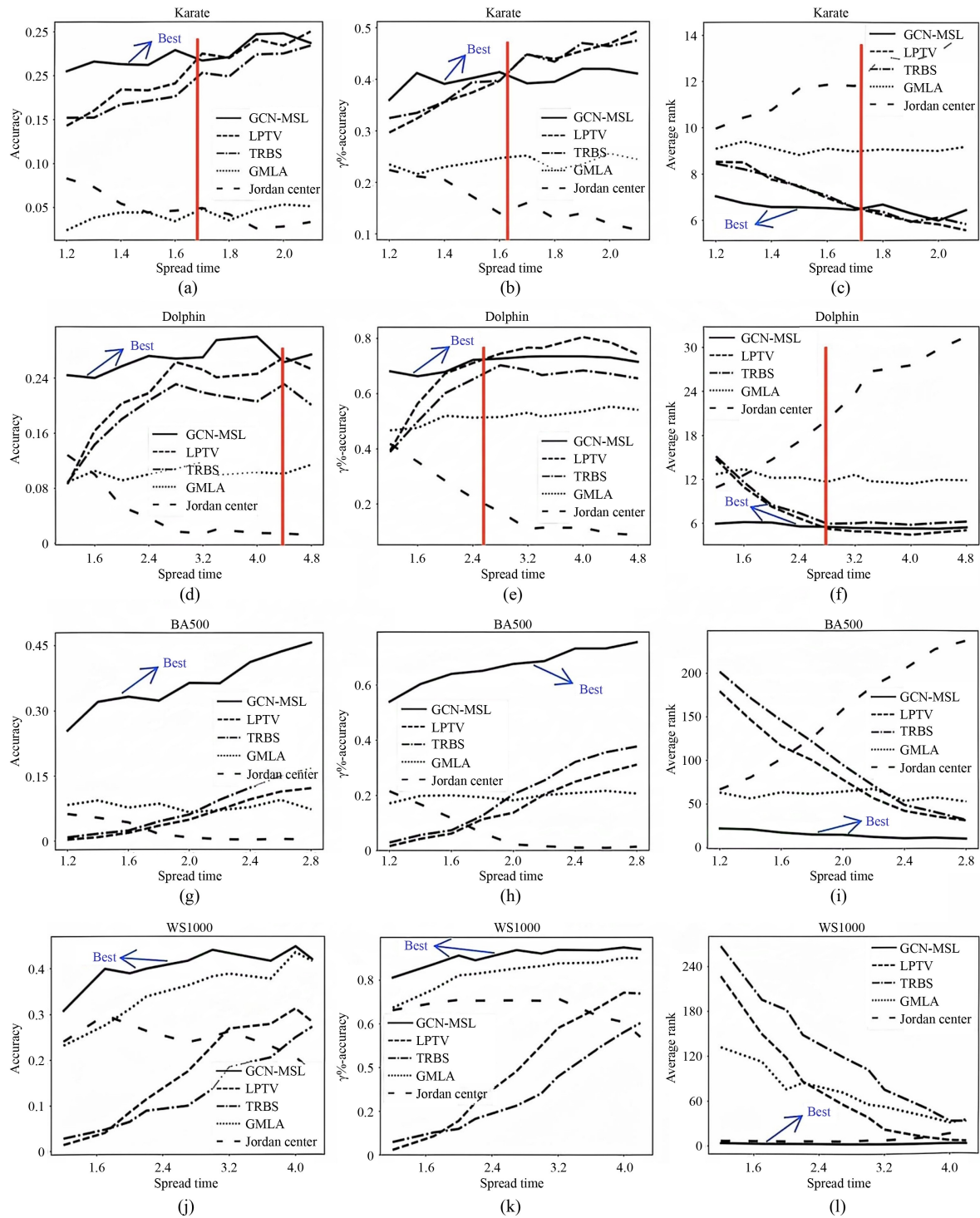


Fig. 6 Results of altering the layout of monitors. (a)–(c): Results in Karate; (d)–(f): results in Dolphin; (g)–(i): results in BA500; (j)–(l): results in WS1000. In each subfigure, the area to the left of the red line indicates the region where GCN-MSL outperforms the other methods. If no red line is present, it indicates that GCN-MSL consistently maintains an advantage.

stages and facilitating timely intervention to prevent the spread of harmful information.

While the proposed method demonstrates significant effectiveness in source localization, it still has certain limitations. First, our approach is limited to addressing

the source localization problem within networks with static topologies. Second, in our simulations of information diffusion, we have solely considered the snowballing diffusion model. Deploying monitors in real-world networks and applying our method for source localization

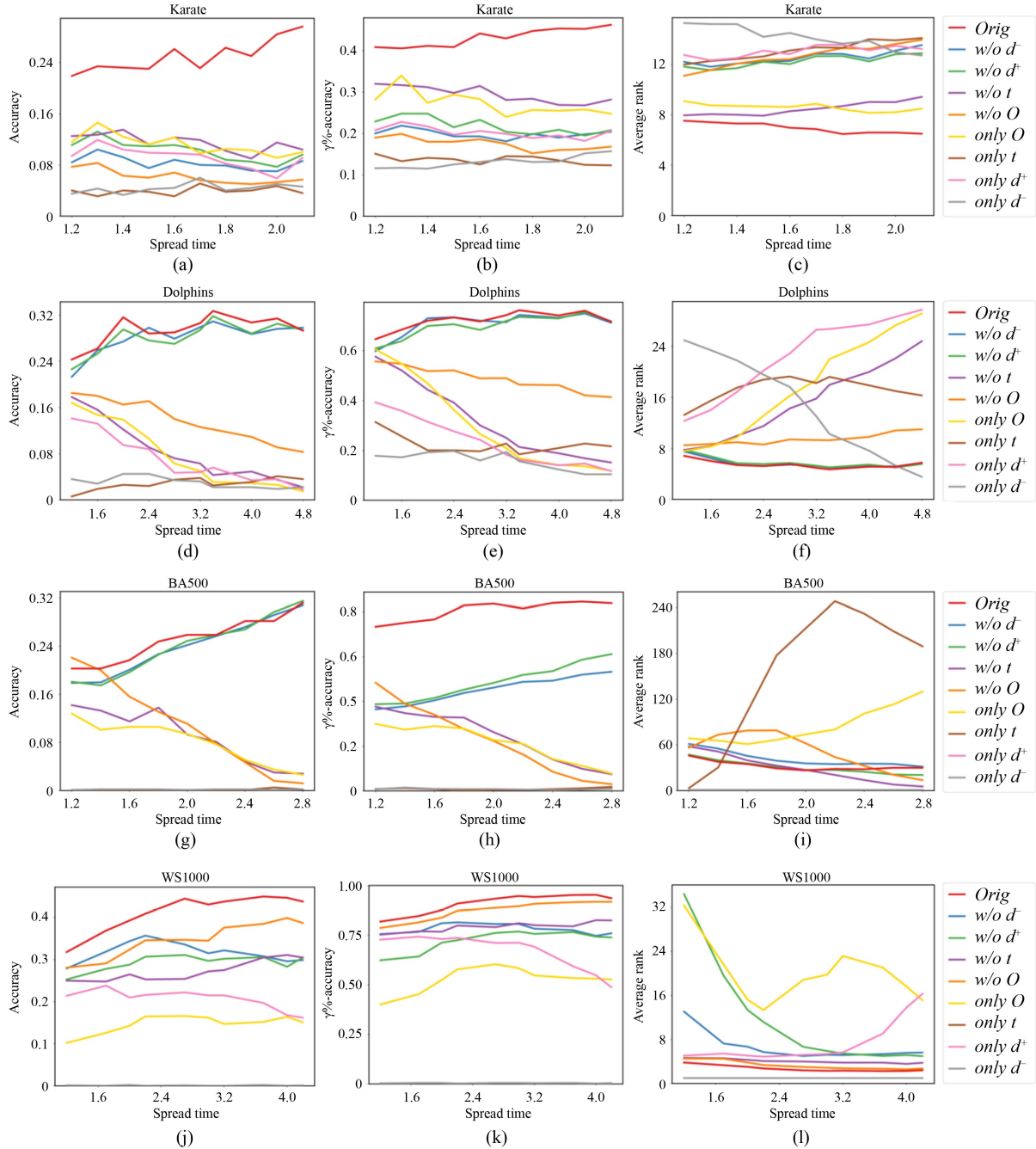


Fig. 7 Results of the ablation study. (a)–(c): Results in Karate; (d)–(f): results in Dolphin; (g)–(i): results in BA500; (j)–(l): results in WS1000. The red curve in each subfigure represents the original model. *Orig* denotes the original model. Model *w/o d⁻* is trained without Feature (4); *w/o d⁺* is trained without Feature (3); *w/o t* is trained without Feature (2); *w/o O* is trained without Feature (1). Model *only O* is trained using only Feature (1); *only t* uses only Feature (2); *only d⁺* uses only Feature (3); *only d⁻* uses only Feature (4).

is highly feasible, though it presents certain challenges. First, issues such as privacy concerns, cost, and network management must be addressed. Secondly, in real-world contexts, network structures are often dynamic, and the challenge lies in how to promptly capture structural changes and adjust the model accordingly. Future work will focus on source localization in networks with dynamically evolving structures, as well as exploring the

impact of network dynamics on localization accuracy. Furthermore, given that information diffusion in the real world occurs through various diffusion models, we intend to extend our method to incorporate a broader spectrum of diffusion models, such as the Independent Cascade (IC) model, the Linear Threshold (LT) model, and the SIR model. This extension will better meet the diverse application demands encountered in practical scenarios.

Competing Interests The authors declare that they have no competing interests.

References

- Bai L, Cui L X, Jiao Y H, Rossi L, Hancock E R (2022). Learning backtrackless aligned-spatial graph convolutional networks for graph classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(2): 783–798
- Cai K C, Xie H, Lui J C S (2018). Information spreading forensics via sequential dependent snapshots. *IEEE/ACM Transactions on Networking*, 26(1): 478–491
- Cheng L, Zhu P C, Gao C, Wang Z, Li X L (2024). A heuristic framework for sources detection in social networks via graph convolutional networks. *IEEE Transactions on Systems, Man, and Cybernetics. Systems*, 54(11): 7002–7014
- Dong M, Zheng B L, Hung N Q V, Su H, Li G H (2019). Multiple rumor source detection with graph convolutional networks. In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. New York: Association for Computing Machinery, 569–578
- Dong M, Zheng B L, Li G H, Li C L, Zheng K, Zhou X F (2022). Wavefront-based multiple rumor sources identification by multi-task learning. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(5): 1068–1078
- Gong C, Li J C, Qian L W, Li S W, Yang Z W, Yang K W (2024). HMSL: Source localization based on higher-order Markov propagation. *Chaos, Solitons, and Fractals*, 182: 114765
- Guo Q, Zhang C, Zhang H S, Fu L Y (2021). IGCN: Infected graph convolutional network based source identification. In: *Proceedings of 2021 IEEE Global Communications Conference*. Piscataway: IEEE Press, 1–6
- Guo Z G, Zhang Y F, Liu S C, Wang X V, Wang L H (2023). Exploring self-organization and self-adaption for smart manufacturing complex networks. *Frontiers of Engineering Management*, 10(2): 206–222
- Hou D P, Gao C, Wang Z, Li X Y, Li X L (2024). Random full-order-coverage based rapid source localization with limited observations for large-scale networks. *IEEE Transactions on Network Science and Engineering*, 11(5): 4213–4226
- Jiang Y N, Wang R R, Sun J S, Wang Y S, You H F, Zhang Y (2024a). Rumor localization, detection and prediction in social network. *IEEE Transactions on Computational Social Systems*, 11(3): 3168–3178
- Jiang Z X, Hu Z L, Huang F L (2024b). Source localization in signed networks based on dynamic message passing algorithm. *Chaos, Solitons, and Fractals*, 188: 115532
- Jin R C, Huang Y F, Zhang Z Y, Dai H Y (2023). On the privacy guarantees of gossip protocols in general networks. *IEEE Transactions on Network Science and Engineering*, 10(6): 3114–3130
- Kingma D P, Ba J (2015). ADAM: A method for stochastic optimization. In: *Proceedings of the 3rd International Conference for Learning Representations*. OpenReview
- Kipf T N, Welling M (2017). Semi-supervised classification with graph convolutional networks. In: *Proceedings of the 5th International Conference on Learning Representations*. Available at Open-Review.net
- Li Y F, Jia C Z (2021). An overview of the reliability metrics for power grids and telecommunication networks. *Frontiers of Engineering Management*, 8(4): 531–544
- Ling C, Jiang J J, Wang J X, Liang Z (2022). Source localization of graph diffusion via variational autoencoders for graph inverse problems. In: *Proceedings of 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. New York: Association for Computing Machinery, 1010–1020
- Meel P, Vishwakarma D K (2020). Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities. *Expert Systems with Applications*, 153: 112986
- Michaël D, Xavier B, Pierre V (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Red Hook: Curran Associates Inc, 3844–3852
- Ou Y, Guo Q, Liu J (2022). Identifying spreading influence nodes for social networks. *Frontiers of Engineering Management*, 9(4): 520–549
- Paluch R, Gajewski L G, Hołyst J A, Szymanski B K (2020). Optimizing sensors placement in complex networks for localization of hidden signal source: A review. *Future Generation Computer Systems*, 112: 1070–1092
- Paluch R, Lu X Y, Suchecki K, Szymański B K, Hołyst J A (2018). Fast and accurate detection of spread source in large complex networks. *Scientific Reports*, 8(1): 2508
- Pan C Y, Wang J, Yan D, Zhang C S, Zhang X Z (2024). A fast algorithm for diffusion source localization in large-scale complex networks. *Journal of Complex Networks*, 12(2): cnae014
- Peng S T, Shu X C, Ruan Z Y, Huang Z G, Xuan Q (2022). Classifying multiclass relationships between ASes using graph convolutional network. *Frontiers of Engineering Management*, 9(4): 653–667
- Pinto P C, Thiran P, Vetterli M (2012). Locating the source of diffusion in large-scale networks. *Physical Review Letters*, 109(6): 068702
- Shah D, Zaman T (2011). Rumors in a network: Who's the culprit? *IEEE Transactions on Information Theory*, 57(8): 5163–5181
- Shelke S, Attar V (2019). Source detection of rumor in social network – A review. *Online Social Networks and Media*, 9: 30–42
- Shen Z S, Cao S N, Wang W X, Di Z R, Stanley H E (2016). Locating the source of diffusion in complex networks by time-reversal backward spreading. *Physical Review. E*, 93(3): 032301
- Tang W, Yang H, Pi J X (2024). Dynamics and control strategies for SLBRS model of computer viruses based on complex networks. *International Journal of Intelligent Systems*, 2024(1): 1–16
- Wang H J, Sun K J (2020). Locating source of heterogeneous propagation model by universal algorithm. *Europhysics Letters*, 131(4): 48001
- Wang J X, Jiang J J, Zhao L. (2022) An invertible graph diffusion neural network for source localization. In: *Proceedings of 31st ACM World Wide Web Conference*. New York: Association for Computing Machinery, 1058–1069
- Wang Z, Hou D, Gao C, Li X, Li X (2023). Lightweight source localization for large-scale social networks. In: *Proceedings of WWW'23: The ACM Web Conference 2023*. New York: Association for Computing Machinery, 286–294

- Wang Z, Wang C K, Pei J S, Ye X J (2017). Multiple source detection without knowing the underlying propagation model. In: Proceedings of Thirty-First AAAI Conference on Artificial Intelligence. Menlo Park: AAAI Press, 217–223
- William L H, Rex Y, Jure L (2017). Inductive representation learning on large graphs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. Red Hook: Curran Associates Inc, 1025–1035
- Wu Z, Pan S, Chen F, Long G, Zhang C, Yu P S (2021). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1): 4–24
- Xu X, Qian T, Xiao Z, Zhang N, Wu J, Zhou F (2024). PGSL: A probabilistic graph diffusion model for source localization. *Expert Systems with Applications*, 238: 122028
- Yang F, Yang S H, Peng Y, Yao Y B, Wang Z W, Li H J, Liu J X, Zhang R S, Li C G (2020). Locating the propagation source in complex networks with a direction-induced search based Gaussian estimator. *Knowledge-Based Systems*, 195: 105674
- Zhao J, Cheong K H (2023). Early identification of diffusion source in complex networks with evidence theory. *Information Sciences*, 642: 119061
- Zhou J, Cui G Q, Zhang Z Y, Yang C, Liu Z Y, Sun M S (2019). Graph neural networks: A Review of methods and applications. *Statistics*, 2019: 2
- Zhou J Y, Jiang Y W, Huang B Q (2021). Source identification of infectious diseases in networks via label ranking. *PLoS One*, 16(1): e0245344
- Zhu K, Ying L (2016). Information source detection in the SIR model: a sample-path-based approach. *IEEE/ACM Transactions on Networking*, 24(1): 408–421