

SourceDetMamba: A Graph-aware State Space Model for Source Detection in Sequential Hypergraphs

Le Cheng^{1,2}, Peican Zhu^{1*}, Yangming Guo³, Chao Gao¹, Zhen Wang^{3*} and Keke Tang^{4*}

¹School of Artificial Intelligence, Optics and Electronics, Northwestern Polytechnical University

²School of Computer Science, Northwestern Polytechnical University

³School of Cybersecurity, Northwestern Polytechnical University

⁴Cyberspace Institute of Advanced Technology, Guangzhou University
{ericcan, w-zhen}@nwpu.edu.cn, tangbohutbh@gmail.com

Abstract

Source detection on graphs has demonstrated high efficacy in identifying rumor origins. Despite advances in machine learning-based methods, many fail to capture intrinsic dynamics of rumor propagation. In this work, we present SourceDetMamba: A Graph-aware State Space Model for Source Detection in Sequential Hypergraphs, which harnesses the recent success of the state space model Mamba, known for its superior global modeling capabilities and computational efficiency, to address this challenge. Specifically, we first employ hypergraphs to model high-order interactions within social networks. Subsequently, temporal network snapshots generated during the propagation process are sequentially fed in reverse order into Mamba to infer underlying propagation dynamics. Finally, to empower the sequential model to effectively capture propagation patterns while integrating structural information, we propose a novel graph-aware state update mechanism, wherein the state of each node is propagated and refined by both temporal dependencies and topological context. Extensive evaluations on eight datasets demonstrate that SourceDetMamba consistently outperforms state-of-the-art approaches.

1 Introduction

Source nodes detection on graphs provides a feasible approach to addressing societal challenges, such as rumor source identification, while simultaneously posing fundamental mathematical problems [Shah and Zaman, 2011; Zhu *et al.*, 2022]. To address this issue, early methods such as LPSI [Wang *et al.*, 2017], OJC [Zhu *et al.*, 2017], and MLE [Pinto *et al.*, 2012] utilize source centrality theory [Shah and Zaman, 2011] and maximum likelihood estimation [Yang *et al.*, 2020] to identify propagation sources. More recently, with the advancement of machine learning, particularly GCNs [Kipf and Welling, 2017], researchers embed user information and network topology into optimized models [Dong *et al.*, 2019;

*Corresponding authors.

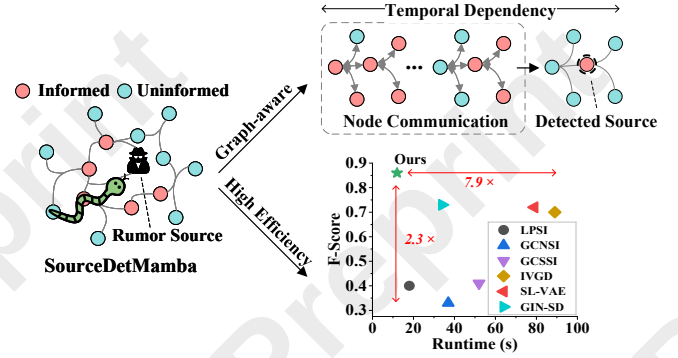


Figure 1: We propose a Mamba-based solution for source detection in sequential hypergraphs, i.e., SourceDetMamba, which demonstrates significant improvements in both accuracy and efficiency compared to baseline methods.

Ling *et al.*, 2022; Wang *et al.*, 2022], continuously achieving state-of-the-art performance across various social network scenarios [Bao *et al.*, 2024; Feng *et al.*, 2024].

However, most of current solutions are based on a single network snapshot. These approaches either heavily rely on handcrafted features for node classification without learning the propagation patterns and trends; or first fit a propagation model to the snapshot before performing source detection. Unfortunately, the process of fitting the propagation model introduces extra errors, hence limiting the subsequent source identification performance. This raises an intuitive question: *can we directly model propagation patterns in reverse order from sequential network snapshots to detect sources?*

Addressing this problem introduces new challenges. First, existing sequence models struggle with this task. Traditional models like RNNs and LSTMs suffer from limited performance due to their inability to handle long-term dependencies effectively [Liang *et al.*, 2024]. While the Transformer architecture improves performance by leveraging attention mechanisms to compute weights for each point in the sequence, its quadratic complexity results in high computational costs for large-scale scenarios [Behrouz and Hashemi, 2024]. Second, directly feeding the features of all nodes in a single network snapshot as a sequence element into the sequential model results in the loss of critical topological information.

We note that state space models (SSMs) have recently made remarkable progress. For instance, the structured state space model (S4) enhances the ability to handle long-range dependencies, achieving powerful representation capabilities with only linear complexity [Gu *et al.*, 2022; Gu *et al.*, 2021]. Building on S4, another pioneer that has garnered significant attention, Mamba, a selective SSM, introduces time-varying parameters to enable content-aware reasoning [Gu and Dao, 2023]. Moreover, Mamba employs a hardware-aware algorithm that allows parallel computation, ensuring efficient training and inference. It has demonstrated impressive performance, matching transformer-level results while maintaining linear computational complexity. However, directly applying Mamba to source detection via sequential snapshots yields suboptimal results. The primary challenge is how to ensure that Mamba effectively incorporates topological information from each network snapshot while learning propagation patterns. Addressing this issue is critical before deploying Mamba for effective source detection.

In this paper, we propose SourceDetMamba: a graph-aware state space model for source detection in sequential hypergraphs. First, hypergraphs are utilized to model the higher-order interactions in real-world networks, such as group chats on social media [Yin *et al.*, 2022; Jiao *et al.*, 2024]. Next, to directly learn the rumor propagation patterns, multiple sequential network snapshots generated during the rumor propagation process are fed into Mamba in a reverse chronological order to capture the underlying dynamics. Finally, to ensure that both the sequential information and the topological structure of each snapshot are effectively incorporated, we innovatively introduce a graph-aware state update mechanism, where each node’s state is determined by both the sequential input and its neighborhood. We evaluate our approach on eight public datasets. Extensive experiments demonstrate that our method outperforms SOTA approaches, establishing new benchmarks in source detection tasks.

Overall, our contributions are summarized as follows:

- We design a unified framework, SourceDetMamba, which leverages the recent advances in state space models to directly learn propagation patterns from reverse-ordered network snapshots.
- We propose a novel graph-aware state update mechanism that integrates sequential propagation dynamics with the underlying network topology, enabling state space models to be effectively applied to sequential tasks on graphs.
- We show by comprehensive experiments on eight real-world datasets that SourceDetMamba consistently outperforms state-of-the-art methods in terms of both accuracy and efficiency.

2 Related Work

2.1 Snapshot-based Source Detection

Network snapshots have become the preferred foundation for source detection methods in recent years due to the convenience and capability to capture user states and network

topology [Cheng *et al.*, 2024b]. Based on source centrality theory [Prakash *et al.*, 2012; Shah and Zaman, 2011], LPSI identifies locally prominent nodes via label propagation [Wang *et al.*, 2017], EPA iteratively infers infection times [Ali *et al.*, 2019], and OJC optimizes Jordan centrality [Zhu *et al.*, 2017]. Although computationally cheap, these heuristics struggle with the complex attributes and diverse dynamics of real social networks [Cheng *et al.*, 2024a]. Machine learning offers a promising alternative [Pei *et al.*, 2024; Pei *et al.*, 2020]. Utilizing GCNs, GCNSI [Dong *et al.*, 2019] and SIGN [Li *et al.*, 2021] incorporate LPSI as a preprocessing step and employ user states for node classification, while GCSSI targets the wavefront of infection [Dong *et al.*, 2022]. From a model architecture perspective, ResGCN adds residual links to deepen message passing [Shah *et al.*, 2020]. Nevertheless, these approaches still under-learn intrinsic diffusion patterns. To address this limitation, IVGD [Wang *et al.*, 2022] and SL-VAE [Ling *et al.*, 2022] model heterogeneity via graph diffusion. However, the extra error in fitting dynamics limits accuracy, and heavy attention modules further raise computation.

2.2 State Space Models

Recently, state space models (SSMs) have emerged as a promising paradigm for sequence modeling, offering an efficient alternative to CNNs and Transformers [Gu *et al.*, 2021]. For instance, Structured State Space for Sequences (S4) leverages the HIPPO matrix to construct a state representation that effectively captures recent inputs while gradually decaying older information [Gu *et al.*, 2020]. This design allows S4 to process long sequences with linear complexity. Building on S4, Mamba introduces adaptive parameters, enabling dynamic filtering of relevant and irrelevant information. Additionally, its hardware-aware design supports efficient storage and computation through parallel scanning, achieving Transformer-level performance while maintaining linear complexity [Gu and Dao, 2023]. In graph-related tasks, Graph Mamba extends Mamba framework to single-graph by incorporating neighborhood tokenization, token ordering, and a selective SSM encoder [Behrouz and Hashemi, 2024]. However, the exploration of state space models on sequential graphs remains an open challenge, leaving a gap in methods designed for such tasks.

In this work, we focus on proposing a state space model for source detection via sequential hypergraph snapshots, which effectively captures propagation patterns through incorporating the topological information of each snapshot.

3 Problem Formulation

Preliminary on Hypergraphs. Hypergraphs extend the pairwise graph structure by accommodating higher-order interactions through hyperedges, enabling the representation of complex relationships in social networks. Formally, a social network can be abstracted as a hypergraph $G = (V, E, \Omega)$, where $V = \{v_1, v_2, \dots, v_n\}$ denotes the set of nodes, and $E = \{e_1, e_2, \dots, e_m\}$ represents the set of hyperedges, with each hyperedge $e_i = \{v_{i1}, v_{i2}, \dots, v_{ij}\}$ consists of a subset of nodes, where $i \in \{1, 2, \dots, m\}$ and $j \in \{1, 2, \dots, n\}$. $\Omega \in \mathbb{R}^{m \times m}$

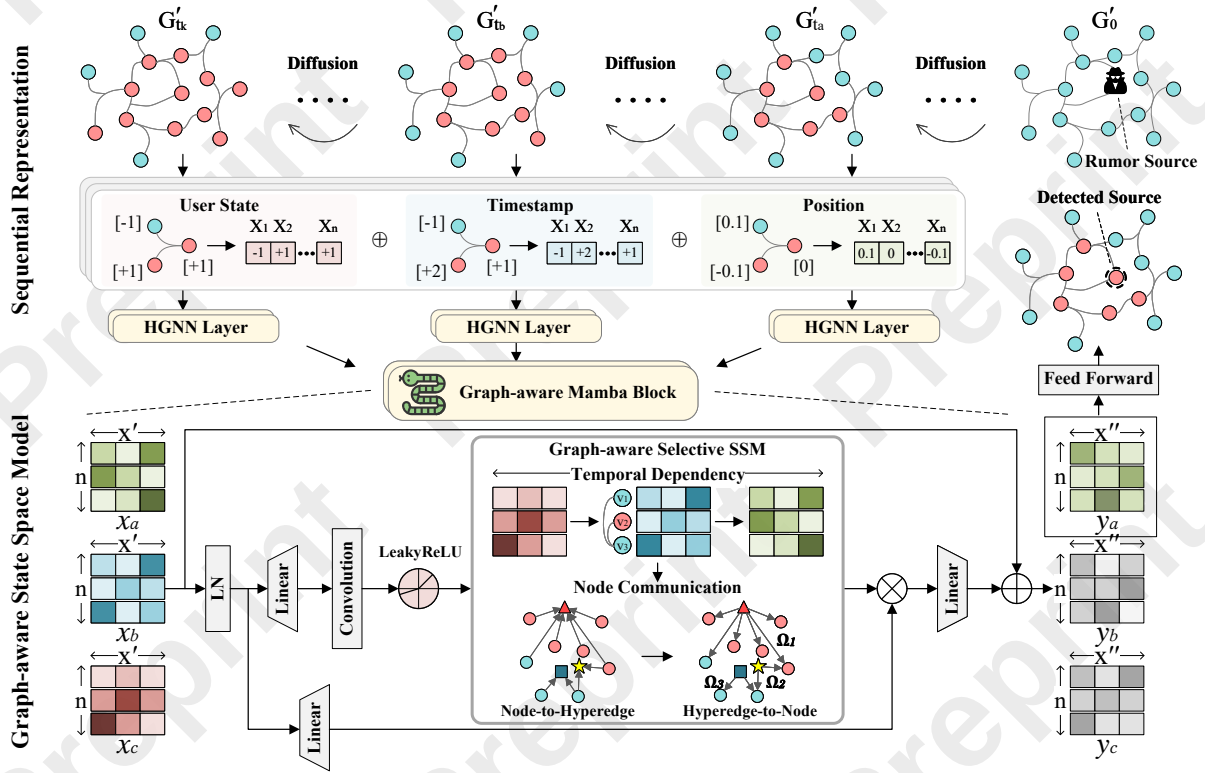


Figure 2: Framework of SourceDetMamba. Starting with hypergraph propagation snapshots, node features are embedded and refined via hypergraph neural networks to capture high-order interactions, which are then input into the graph-aware state space model in a reverse order to learn propagation patterns. Finally, the source nodes are identified based on the last element of the output sequence.

is a diagonal matrix, where each element corresponds to the weight of a hyperedge. Given the relationship between nodes and hyperedges, the hypergraph G can be represented by an incidence matrix $H \in \mathbb{R}^{n \times m}$, each entry H_{ve} indicates the participation of a node v in a hyperedge e :

$$H_{ve} = \begin{cases} 1, & v \in e \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Consequently, the degree of nodes and hyperedges are expressed using diagonal matrices $D_V \in \mathbb{R}^{n \times n}$ and $D_E \in \mathbb{R}^{m \times m}$, where each entry $D_{vv} = \sum_{e=1}^m H_{ve}$ and $D_{ee} = \sum_{v=1}^n H_{ve}$, respectively.

Propagation Process on Hypergraphs. The rumor propagation process in social networks evolves over time t , where the initially uninformed source nodes transition to the informed state at $t = 0$, triggering the spread of information. Unlike the pairwise interactions modeled in conventional graphs, propagation on hypergraphs encompasses both low-order interactions between individual nodes and high-order interactions within hyperedges, representing peer influence or collective pressure among groups. During the low-order interaction phase, an informed node v_i disseminates information to its neighbors with a probability p_i . In the high-order interaction phase, a node transitions from the uninformed to the informed state if the cumulative influence from its neighbors exceeds a threshold p_Δ . Several classical models, such as SI, SIR, IC, and LT [Battiston *et al.*, 2020;

de Arruda *et al.*, 2020], have been proposed to simulate the propagation process on hypergraphs.

In summary, the propagation on hypergraphs can be represented as a time series $\{G'(t), t \geq 0\}$, where $G'(t)$ is the network snapshot at time t , encapsulating information such as network structure, node states, and propagation information.

Source Detection in Sequential Hypergraphs. As the propagation evolves, a series of network snapshots are captured at distinct time points $T = \{t_a, t_b, \dots, t_k\}$, which are not necessarily continuous or evenly spaced. Consequently, source detection on sequential hypergraphs can be defined as:

$$\hat{s} = f(\{G'(t_i)\}_{t_i \in T}), \quad (2)$$

where $f(\cdot)$ denotes the source detection method, and $\hat{s}$ represents the identified source set.

Main Challenges of Rumor Source Detection in Sequential Hypergraphs.

- (1) **Modeling Sequential Hypergraphs and Node Feature Embedding.** Effectively modeling sequential hypergraphs while embedding node features to capture high-order interactions remains a formidable challenge.
- (2) **Integration of Structural Properties into Node Feature Updates.** Although Mamba has demonstrated strong sequential modeling capabilities, directly feeding node features from a snapshot as a sequence element results in the loss of critical topological information.

4 Method

We propose SourceDetMamba, a framework explicitly designed to address the challenges above. Specifically, we introduce the **Sequential Representation and Feature Fusion** module for Challenge (1), and design a **Graph-aware State Space Model** to handle Challenge (2).

4.1 Sequential Representation and Feature Fusion

Throughout the propagation process, network snapshots are captured at discrete time points $T = \{t_a, t_b, \dots, t_k\}$. Within each snapshot, nodes are classified into two subgraphs: $G_{t_i}^+$ (Informed) and $G_{t_i}^-$ (Uninformed). To maximize the utilization of essential information and reinforce comparative features, we embed key attributes, including user states, diffusion timestamps, and positional information, into the initial feature representations.

User State Information X_i^1 . The state of each node is defined by its participation in the rumor propagation process. Specifically, node $v_i \in G_{t_i}^+$ if it has informed the rumor; otherwise, $v_i \in G_{t_i}^-$. Accordingly, X_i^1 is expressed as:

$$X_i^1 = \begin{cases} +1, & v_i \in G_{t_i}^+ \\ -1, & \text{otherwise.} \end{cases} \quad (3)$$

Diffusion Information X_i^2 . Social platforms like Facebook and Twitter attach timestamps to user posts, providing critical information for source detection. Hence, diffusion information X_i^2 is defined as:

$$X_i^2 = \begin{cases} t_{v_i}, & v_i \in G_{t_i}^+ \\ -1, & \text{otherwise.} \end{cases} \quad (4)$$

Positional Information X_i^3 . Existing GCN models learn structural node information with invariant node positions [Srinivasan and Ribeiro, 2019]. However, relative positions between nodes play a critical role in global message propagation. Leveraging the generative capacity of Laplacian positional encodings (PEs) [Dwivedi *et al.*, 2020], we incorporate it as the positional feature of nodes. Inspired by source centrality theory, we focus on calculating PEs for the infected subgraph $G_{t_i}^+$ rather than the entire graph, highlighting the positional significance of sources. The symmetric normalized Laplacian matrix for $G_{t_i}^+$ is defined as:

$$L_+^{sym} = I - D_{V+}^{-\frac{1}{2}} H_+ D_{E+}^{-1} H_+^T D_{V+}^{-\frac{1}{2}}, \quad (5)$$

where I denotes the identity matrix, H_+ is the incidence matrix of the subgraph $G_{t_i}^+$, D_{V+} and D_{E+} represent the degree matrices of nodes and hyperedges, respectively. The Laplacian matrix L_+^{sym} can further be decomposed as:

$$\Delta L_+^{sym} = \Gamma \Lambda \Gamma^T, \quad (6)$$

where Γ consists of eigenvectors and Λ is the diagonal matrix of eigenvalues. The eigenvectors associated with the smallest non-zero eigenvalues provide the positional encodings, which we adopt as the node-level positional feature X_i^3 :

$$X_i^3 = \begin{cases} \Gamma_i, & v_i \in G_{t_i}^+ \\ -1, & \text{otherwise.} \end{cases} \quad (7)$$

Finally, the initial feature of node v_i is obtained by concatenating the individual components:

$$X_i = [\|_{x=1}^3 X_i^x]. \quad (8)$$

Sequential Feature Preprocessing. For the derived sequential hypergraph features $\{X(t_i)\}_{t_i \in T}$. Unlike the feature aggregation process in pairwise graphs, hypergraphs require a unique approach due to the role of hyperedges as mediums for interactions among multiple nodes. To address this, we employ the Hypergraph Neural Network (HGNN) [Feng *et al.*, 2019; Gao *et al.*, 2022; Bai *et al.*, 2021] that aggregates features in two stages: node-to-hyperedge and hyperedge-to-node. The process is formulated as:

$$X^{(l+1)} = \sigma \left(D_V^{-1/2} H \Omega D_E^{-1} H^T D_V^{-1/2} X^{(l)} W^{(l)} \right), \quad (9)$$

where $X^{(l)}$ represents the node features at layer l , H is the incidence matrix, D_V and D_E are the diagonal degree matrices of nodes and hyperedges, respectively. Ω denotes the hyperedge weight matrix, and $W^{(l)}$ is the trainable parameters. The activation function $\sigma(\cdot)$ introduces non-linearity, enabling the model to capture the intricate high-order interactions effectively. Through Eq. (9), the initial feature sequence $\{X(t_i)^{n \times x}\}_{t_i \in T}$ is projected into $\{X'(t_i)^{n \times x'}\}_{t_i \in T}$, serving as the foundation for downstream tasks.

4.2 Graph-aware State Space Model

State Space Model. By capturing sequential dynamics through explicitly maintaining a latent state that evolves over time, State Space Models (SSMs) map an input sequence $x_t \in \mathbb{R}$ to a latent state representation $h_t \in \mathbb{R}^N$, and subsequently derive a predicted output sequence $y_t \in \mathbb{R}$. While continuous SSMs provide a mathematically elegant framework, finding an analytical solution for the state representation h_t remains a significant challenge. Moreover, the majority of practical sequences, such as textual data, are inherently discrete. Consequently, discrete SSMs are adopted with four parameters (Δ, A, B, C), the sequence-to-sequence transformation can be expressed as:

$$\begin{aligned} h_t &= \bar{A} h_{t-1} + \bar{B} x_t \\ y_t &= C h_t + D x_t, \end{aligned} \quad (10)$$

where C and D represent projection parameters mapping the latent state and input to the output space. $\bar{A}$ and $\bar{B}$ are derived through the zero-order hold (ZOH) discretization method with a sampling step size Δ . Specifically:

$$\begin{aligned} \bar{A} &= \exp(\Delta A) \\ \bar{B} &= (\Delta A)^{-1} (\exp(\Delta A) - I) \cdot \Delta B. \end{aligned} \quad (11)$$

This discretization strategy ensures both numerical stability and computational efficiency. However, the parameters A, B, C, Δ are fixed across all time steps, adhering to the Linear Time Invariant (LTI) property, which restricts the model's ability to accommodate dynamic variations in input sequences. To address this limitation, Mamba introduces a content-aware reasoning framework by modeling B, C, Δ as functions of the input. This transformation results in a selective SSM capable of adapting to input-dependent variations.

We feed the sequence of hypergraph features $\{X'(t_i)\}$ into several Mamba blocks in reverse temporal order to capture the underlying propagation dynamics. Moreover, by

utilizing convolution kernel $\bar{K} = (\bar{CB}, \bar{CAB}, \dots, \bar{CA}^k \bar{B}, \dots)$, we achieve the sequence-to-sequence transformation through $y = X' * \bar{K}$ with linear time complexity.

Graph-aware State Update Mechanism. While Mamba enables input-dependent learning with linear time complexity, directly feeding hypergraph snapshots as individual sequence elements into the model risks losing essential topological information. To address this, we propose a graph-aware state update mechanism allows for the update of node features by enabling node communication, where the state of each node is determined collaboratively by both the sequence input and the states of its neighbors, as expressed by:

$$h_t = \bar{A}h_{t-1} + \bar{B}x_t + h_N. \quad (12)$$

Similar to the convolutional process on hypergraphs, the neighbor state h_N is updated in two steps: node-to-hyperedge and hyperedge-to-node.

(1) Node-to-Hyperedge Acting as mediators between nodes, hyperedges first collect the averaged state information from its connected nodes:

$$h_{edge} = H^T D_V^{-1} h_{t-1}. \quad (13)$$

(2) Hyperedge-to-Node In this process, the state information is aggregated back to the nodes from the hyperedges:

$$h_N = HD_E^{-1} h_{edge}. \quad (14)$$

Through aggregating the state information of neighboring nodes, the SSM acquires the capability to perceive the underlying graph structure. However, in the context of rumor propagation, distinct groups of nodes contribute differently to the spread of information. Therefore, hyperedges should be weighted based on their respective states. Given that attention mechanisms introduce additional computational complexity, we utilize a MLP block with an activation function and sigmoid to learn the weights of each hyperedge:

$$\Omega_e = \text{sigmoid}(\sigma(\text{MLP}(h_{t-1}))). \quad (15)$$

Consequently, the state update process is summarized as:

$$h_t = \bar{A}h_{t-1} + \bar{B}x_t + HD_E^{-1} \Omega H^T D_V^{-1} h_{t-1}. \quad (16)$$

The proposed graph-aware state update mechanism ensures that the node feature updates are informed by both the sequential data and the structural context of the hypergraph. Moreover, the adoption of hyperedge weights enables the model to dynamically adjust the influence of different hyperedges based on the propagation dynamics, providing a more adaptive and context-sensitive approach to learning rumor propagation patterns.

4.3 Optimization and Training

For the output sequence y , we focus on the last element, i.e., y_1 , for source detection task. To alleviate the class imbalance resulting from the substantial disparity between the sample numbers of source and non-source nodes, we introduce a parameter ξ to mitigate the prediction bias:

$$\xi = \frac{|s|}{n - |s|}, \quad (17)$$

where n denotes the number of nodes, and $|s|$ represents the number of sources. This harmonic parameter equalizes the contribution of all samples. Consequently, the optimization objective is to minimize the discrepancy between the predicted set of sources and the actual sources, formalized as:

$$\min_G \text{Loss} = \sum_{v_i \in s} \mathcal{L}_i + \xi \sum_{v_j \in (V-s)} \mathcal{L}_j + \lambda \|w\|_2, \quad (18)$$

where $\mathcal{L}$ denotes the cross-entropy loss, and for a sample x with corresponding label z , $\mathcal{L}(x, z) = -\log(x) \times z$. The final term represents the L_2 regularization of the weight vector W , with λ controlling the regularization strength.

5 Experiments

5.1 Experimental Settings

Datasets. Eight datasets with varying scales and characteristics are used to evaluate the performance of our SourceDet-Mamba. Considered datasets includes Zoo [Asuncion *et al.*, 2007], House [Chodrow *et al.*, 2021], NTU2012 [Chen *et al.*, 2003], Mushroom [Asuncion *et al.*, 2007], ModelNet40 [Wu *et al.*, 2015], 20News [Asuncion *et al.*, 2007], PubMed [Yadati *et al.*, 2019] and Walmart [Amburg *et al.*, 2020].

Baselines. Different categories of methods are selected as baselines, including source centrality-based methods such as LPSI [Wang *et al.*, 2017] and EPA [Ali *et al.*, 2019], user state-based methods like GCNSI [Dong *et al.*, 2019], SIGN [Li *et al.*, 2021], GCSSI [Dong *et al.*, 2022] and ResGCN [Shah *et al.*, 2020], and propagation information-integrated methods such as IVGD [Wang *et al.*, 2022], SL-VAE [Ling *et al.*, 2022], and GIN-SD [Cheng *et al.*, 2024b].

Implementation. Given the independence of users on social platforms and the short-term nature of rumor propagation, we simulate the rumor spread using a heterogeneous independent cascade model. The propagation process is triggered by randomly selecting 5% of the nodes as sources. For low-order interactions, the propagation rate p_i of each node follows a uniform distribution $U(0, 0.5)$. In the high-order interaction phase, the state transition probability p_Δ for each node in a hyperedge is proportional to the number of informed nodes, formulated as $p_\Delta = 0.3 \cdot (|e \cap v^+|/|e|)$, $v^+ \in G_{t_i}^+$. The sequential snapshots are captured when 10%, 20%, and 30% of the nodes in the network are informed, with the resulting dataset is split into training and testing sets in an 8:2 ratio. For the training process, the model's learning rate and weight decay are set to 10^{-3} and 10^{-5} , respectively. In the state space model, each input in the sequence undergoes a state update via a two-layer stack, with state size equals to 128. Hyperedge weights are calculated through two linear layers, each followed by an activation function, and the state is mapped to a single scalar value using a sigmoid function. The hypergraph snapshots are first subjected to clique expansion before being applied to the baseline methods. All experiments are conducted on a workstation equipped with four NVIDIA RTX 3090Ti GPUs.

Metrics. ACC (Accuracy), F-Score, and AUC (Area Under the Curve) are selected as evaluation metrics to assess the

Methods	Zoo			House			NTU2012			Mushroom		
	ACC	F-Score	AUC	ACC	F-Score	AUC	ACC	F-Score	AUC	ACC	F-Score	AUC
LPSI	0.802	0.345	0.779	0.812	0.347	0.807	0.787	0.321	0.821	0.815	0.329	0.775
EPA	0.789	0.351	0.785	0.814	0.340	0.811	0.794	0.312	0.820	0.817	0.326	0.774
GCNSI	0.823	0.372	0.810	0.828	0.329	0.811	0.817	0.306	0.825	0.818	0.362	0.811
SIGN	0.821	0.442	0.819	0.826	0.426	0.814	0.803	0.398	0.813	0.814	0.415	0.824
GCSSI	0.811	0.415	0.807	0.826	0.412	0.814	0.805	0.389	0.799	0.805	0.389	0.816
ResGCN	0.819	0.503	0.816	0.825	0.467	0.811	0.807	0.442	0.819	0.829	0.458	0.817
IVGD	0.847	0.589	0.835	0.824	0.529	0.836	0.859	0.532	0.856	0.851	0.526	0.864
SL-VAE	0.851	0.602	0.830	0.842	0.531	0.824	0.846	0.529	0.839	0.864	0.547	0.867
GIN-SD	0.861	0.694	0.834	0.859	0.624	0.854	0.860	0.634	0.841	0.874	0.642	0.884
Ours	0.915	0.797	0.920	0.938	0.836	0.941	0.947	0.869	0.944	0.929	0.827	0.930

Methods	ModelNet40			20News			PubMed			Walmart		
	ACC	F-Score	AUC	ACC	F-Score	AUC	ACC	F-Score	AUC	ACC	F-Score	AUC
LPSI	0.784	0.254	0.771	0.745	0.273	0.741	0.759	0.215	0.764	0.718	0.112	0.718
EPA	0.791	0.258	0.797	0.771	0.284	0.767	0.774	0.205	0.794	0.715	0.108	0.730
GCNSI	0.814	0.278	0.816	0.805	0.237	0.801	0.798	0.202	0.791	0.808	0.174	0.810
SIGN	0.824	0.442	0.821	0.814	0.364	0.806	0.794	0.254	0.791	0.816	0.234	0.815
GCSSI	0.811	0.384	0.795	0.817	0.379	0.805	0.814	0.203	0.807	0.811	0.206	0.806
ResGCN	0.820	0.497	0.819	0.821	0.440	0.807	0.816	0.279	0.823	0.824	0.251	0.823
IVGD	0.820	0.629	0.827	0.815	0.557	0.823	0.819	0.563	0.817	0.810	0.547	0.823
SL-VAE	0.842	0.618	0.834	0.829	0.587	0.822	0.824	0.557	0.824	0.824	0.525	0.830
GIN-SD	0.836	0.612	0.824	0.831	0.584	0.819	0.822	0.587	0.828	0.831	0.536	0.825
Ours	0.921	0.814	0.926	0.908	0.792	0.918	0.905	0.784	0.897	0.879	0.715	0.865

Table 1: The performance of source detection across all datasets for each method, with the best results being highlighted in bold.

performance of different methods. ACC measures the proportion of correctly classified samples. F-Score comprises *Precision* and *Recall*, where $Precision = |\hat{s} \cap s| / |\hat{s}|$ represents the proportion of true sources in the detected source set, and $Recall = |\hat{s} \cap s| / |s|$ indicates the proportion of true sources detected. The AUC evaluates the model’s ability to differentiate between source and non-source nodes across varying thresholds. These metrics provide a comprehensive evaluation of the performance and robustness of source detection methods.

5.2 Performance Analysis

Comparison with State-of-the-art Methods. Table 1 presents the comparative results of our proposed SourceDetMamba against baseline methods, revealing several key insights. First, all methods achieve higher ACC than F-Score, primarily due to the imbalance between the large number of non-source nodes and the relatively small number of source nodes. As ACC measures the overall proportion of correctly classified samples, this disparity highlights the impact of class imbalance, particularly for methods such as LPSI and EPA based on source centrality, and machine learning approaches like GCNSI, SIGN, and GCSSI. Second, methods relying solely on source centrality demonstrate lower F-Score due to the inherent randomness of the propagation process, which limits the effectiveness of topology-only approaches. Machine learning-based methods that incorporate user features perform better, and methods that model the propagation process, such as IVGD and SL-VAE, or integrate propagation features directly, like GIN-SD, achieve even greater performance improvements. Notably, SourceDetMamba outperforms all baselines, delivering 10%-23% higher accuracy than propagation-aware methods, 25%-35% higher than user state-based methods, and even doubling the performance of

source centrality-based approaches. These significant improvements stem from SourceDetMamba’s ability to directly learn propagation dynamics through Mamba blocks and its graph-aware state update mechanism, which seamlessly incorporates hypergraph topology into the learning process.

Visualization. To provide an intuitive illustration of the source detection results, we visualize the outputs of SourceDetMamba alongside those of representative methods on the NTU2012 dataset [Chen *et al.*, 2003] in Fig. 3. As compared to baselines, our approach demonstrates better performance in identifying the sources.

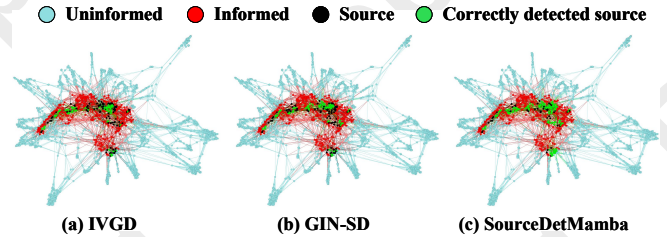


Figure 3: Visualization of source detection results on NTU2012.

Computational Efficiency. To evaluate computational efficiency, we compare the runtime and accuracy of the proposed method against baseline approaches, as shown in Fig. 4. The results reveal that methods not considering propagation dynamics achieve faster runtime but suffer from lower F-scores, indicating limited accuracy. Conversely, methods incorporating propagation dynamics demonstrate improved accuracy at the expense of significantly higher computational complexity. In contrast, our proposed method achieves a better balance, excelling in both accuracy and runtime efficiency. These findings validate the effectiveness of SourceDetMamba in deliv-

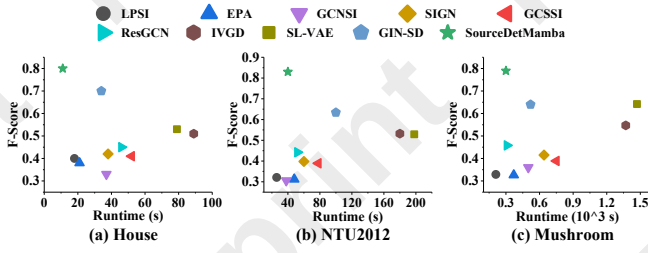


Figure 4: Comparison of runtime and F-Score across baseline methods, highlighting the balance achieved by SourceDetMamba.

ering SOTA performance with computational efficiency.

5.3 Ablation Study and Other Analyses

Effects of Sequential Snapshots and Positional Encoding.

We first remove the sequential snapshots, retaining only the earliest one. The results in Table 2 show a noticeable performance decline compared to the complete model, highlighting the importance of utilizing multiple snapshots to capture propagation patterns. Next, we exclude the positional encodings in w/o PEs, the observed performance degradation highlights the critical role that PEs play in capturing the structural context and enhancing the model’s robustness.

Effects of Exiting Sequential Models. We employ transformation and attention mechanisms to facilitate reverse learning of propagation patterns in w/ Attention, the results demonstrate a noticeable decline compared to SourceDetMamba. When sequential features are directly fed into an LSTM, the performance is similar to that of the attention-based model. In the w/ Att & LSTM, we first apply attention fusion to the features and then input them into the LSTM. However, the results reveal a decrease in performance compared to using attention or LSTM individually. The performance is mainly attributed to the challenge of existing sequential models in effectively learning input dependencies.

Effects of the Graph-aware State Update Mechanism.

By removing node communication from Mamba and relying solely on the temporal states of individual nodes for prediction, we observe a significant performance drop. This highlights the critical role of the graph-aware state update mechanism in sequential graph tasks. Additionally, when hyperedge weight computation is omitted, performance improves compared to the model without the graph-aware mechanism, yet it still falls short of SourceDetMamba. These results confirm the effectiveness and validity of integrating topological structure and differentiating hyperedges in our proposed model.

Methods	House	NTU2012	20News	PubMed
w/ Single snapshot	0.758	0.768	0.694	0.672
w/o PEs	0.822	0.854	0.780	0.769
w/ Attention	0.798	0.805	0.703	0.712
w/ LSTM	0.785	0.794	0.691	0.716
w/ Att & LSTM	0.782	0.806	0.683	0.714
w/o Graph	0.790	0.804	0.735	0.742
w/o Edge weights	0.824	0.847	0.772	0.759
SourceDetMamba	0.836	0.869	0.792	0.784

Table 2: Performance of different SourceDetMamba variants.

Models	House		ModelNet40		PubMed	
	ACC	F-Score	ACC	F-Score	ACC	F-Score
SI	0.925	0.825	0.917	0.816	0.910	0.774
SIS	0.926	0.804	0.918	0.794	0.915	0.754
SIR	0.921	0.794	0.912	0.785	0.921	0.742
IC	0.938	0.836	0.921	0.814	0.905	0.784

Table 3: Performance of SourceDetMamba under different information diffusion models.

Impact of Information Diffusion Models. To address diverse scenarios, we employ various information diffusion models and summarize the performance of SourceDetMamba in Table 3. The results demonstrate that SourceDetMamba consistently performs well across all tested models. A slight decline is observed for the SIS and SIR models due to the possibility of sources recovery, which introduces ambiguity into the detection process. Nonetheless, the strong overall performance underscores the robustness and adaptability of SourceDetMamba, reinforcing its effectiveness across a range of information propagation dynamics.

Impact of Propagation Scale and Snapshot Intervals.

We set the initial snapshot sizes to 10%, 20%, and 30%, and vary the interval between subsequent snapshots from 5% to 25% with an increment of 5%. The results of SourceDetMamba are shown in Fig. 5. Firstly, across all datasets, the F-Score decreases as the propagation scale increases. This is primarily due to the necessity of identifying a larger number of samples, which leads to increased errors. Secondly, varying the snapshot intervals highlights that overly close snapshots may fail to capture sufficient propagation dynamics, while excessively distant snapshots may over-represent the influence of non-source nodes. Therefore, an appropriate snapshot interval is beneficial for effective source detection.

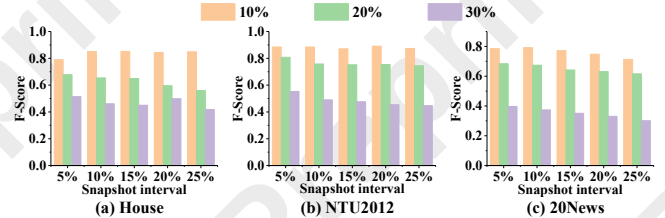


Figure 5: Performance of SourceDetMamba under different propagation scales and snapshot intervals.

6 Conclusion

This paper presents a new perspective on rumor source detection by focusing on effective learning in sequential hypergraphs. The key idea involves reverse-order input of hypergraph snapshots into the Mamba state space model to learn propagation patterns, followed by the proposal of a graph-aware state update mechanism, which integrates topological information with temporal dependencies. Extensive experiments demonstrate that this approach significantly outperforms existing methods in sequential snapshot-based rumor source detection, providing a robust solution for sequential network learning. We hope that this work will inspire further research into topological-aware models in this field.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Grant Nos. U22B2036, 62025602, 62471403, 62073263, 62472117), the Fundamental Research Funds for the Central Universities (Grant Nos. G2024WD0151, D5000240309), the Technological Innovation Team of Shaanxi Province (Grant No. 2025RS-CXTD-009), the International Cooperation Project of Shaanxi Province (Grant No. 2025GH-YBXM-017), the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2025A1515010157), the Science and Technology Projects in Guangzhou (Grant No. 2025A03J0137), the National Foreign Experts Program (Grant No. G2023183019), and the Tencent Foundation and XPLOER PRIZE.

References

- [Ali *et al.*, 2019] Syed Shafat Ali, Tarique Anwar, Ajay Rastogi, and Syed Afzal Murtaza Rizvi. EPA: Exoneration and prominence based age for infection source identification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 891–900, 2019.
- [Amburg *et al.*, 2020] Ilya Amburg, Nate Veldt, and Austin Benson. Clustering in graphs and hypergraphs with categorical edge labels. In *Proceedings of The Web Conference 2020*, pages 706–717, 2020.
- [Asuncion *et al.*, 2007] Arthur Asuncion, David Newman, et al. Uci machine learning repository, 2007.
- [Bai *et al.*, 2021] Song Bai, Feihu Zhang, and Philip HS Torr. Hypergraph convolution and hypergraph attention. *Pattern Recognition*, 110:107637, 2021.
- [Bao *et al.*, 2024] Qing Bao, Ying Jiang, Wang Zhang, Pengfei Jiao, and Jing Su. Graph contrastive learning for source localization in social networks. *Information Sciences*, page 121090, 2024.
- [Battiston *et al.*, 2020] Federico Battiston, Giulia Cencetti, Iacopo Iacopini, Vito Latora, Maxime Lucas, Alice Pataña, Jean-Gabriel Young, and Giovanni Petri. Networks beyond pairwise interactions: Structure and dynamics. *Physics Reports*, 874:1–92, 2020.
- [Behrouz and Hashemi, 2024] Ali Behrouz and Farnoosh Hashemi. Graph mamba: Towards learning on graphs with state space models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 119–130, 2024.
- [Chen *et al.*, 2003] Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen, and Ming Ouhyoung. On visual similarity based 3d model retrieval. In *Computer graphics forum*, volume 22, pages 223–232. Wiley Online Library, 2003.
- [Cheng *et al.*, 2024a] Le Cheng, Peican Zhu, Chao Gao, Zhen Wang, and Xuelong Li. A heuristic framework for sources detection in social networks via graph convolutional networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2024.
- [Cheng *et al.*, 2024b] Le Cheng, Peican Zhu, Keke Tang, Chao Gao, and Zhen Wang. GIN-SD: source detection in graphs with incomplete nodes via positional encoding and attentive fusion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 55–63, 2024.
- [Chodrow *et al.*, 2021] Philip S Chodrow, Nate Veldt, and Austin R Benson. Generative hypergraph clustering: From blockmodels to modularity. *Science Advances*, 7(28):eabh1303, 2021.
- [de Arruda *et al.*, 2020] Guilherme Ferraz de Arruda, Giovanni Petri, and Yamir Moreno. Social contagion models on hypergraphs. *Physical Review Research*, 2(2):023032, 2020.
- [Dong *et al.*, 2019] Ming Dong, Bolong Zheng, Nguyen Quoc Viet Hung, Han Su, and Guohui Li. Multiple rumor source detection with graph convolutional networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 569–578, 2019.
- [Dong *et al.*, 2022] Ming Dong, Bolong Zheng, Guohui Li, Chenliang Li, Kai Zheng, and Xiaofang Zhou. Wavefront-based multiple rumor sources identification by multi-task learning. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(5):1068–1078, 2022.
- [Dwivedi *et al.*, 2020] Vijay Prakash Dwivedi, Chaitanya K Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982*, 2020.
- [Feng *et al.*, 2019] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3558–3565, 2019.
- [Feng *et al.*, 2024] Bingdao Feng, Di Jin, Xiaobao Wang, Fangyu Cheng, and Siqi Guo. Backdoor attacks on unsupervised graph representation learning. *Neural Networks*, 180:106668, 2024.
- [Gao *et al.*, 2022] Yue Gao, Yifan Feng, Shuyi Ji, and Rongrong Ji. HGNN+: General hypergraph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3181–3199, 2022.
- [Gu and Dao, 2023] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.
- [Gu *et al.*, 2020] Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems*, 33:1474–1487, 2020.
- [Gu *et al.*, 2021] Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585, 2021.
- [Gu *et al.*, 2022] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured

- state spaces. In *The International Conference on Learning Representations (ICLR)*, 2022.
- [Jiao *et al.*, 2024] Pengfei Jiao, Hongqian Chen, Qing Bao, Wang Zhang, and Huaming Wu. Enhancing multi-scale diffusion prediction via sequential hypergraphs and adversarial learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8571–8581, 2024.
- [Kipf and Welling, 2017] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR*, 2017.
- [Li *et al.*, 2021] Liang Li, Jianye Zhou, Yuewen Jiang, and Biqing Huang. Propagation source identification of infectious diseases with graph convolutional networks. *Journal of Biomedical Informatics*, 116:103720, 2021.
- [Liang *et al.*, 2024] Dingkan Liang, Xin Zhou, Wei Xu, Xingkui Zhu, Zhikang Zou, Xiaoqing Ye, Xiao Tan, and Xiang Bai. PointMamba: A simple state space model for point cloud analysis. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [Ling *et al.*, 2022] Chen Ling, Junji Jiang, Junxiang Wang, and Zhao Liang. Source localization of graph diffusion via variational autoencoders for graph inverse problems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1010–1020, 2022.
- [Pei *et al.*, 2020] Hongbin Pei, Bo Yang, Jiming Liu, and Kevin Chen-Chuan Chang. Active surveillance via group sparse bayesian learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1133–1148, 2020.
- [Pei *et al.*, 2024] Hongbin Pei, Yu Li, Huiqi Deng, Jingxin Hai, Pinghui Wang, Jie Ma, Jing Tao, Yuheng Xiong, and Xiaohong Guan. Multi-track message passing: tackling oversmoothing and oversquashing in graph learning via preventing heterophily mixing. In *Forty-first International Conference on Machine Learning*, 2024.
- [Pinto *et al.*, 2012] Pedro C Pinto, Patrick Thiran, and Martin Vetterli. Locating the source of diffusion in large-scale networks. *Physical Review Letters*, 109(6):068702, 2012.
- [Prakash *et al.*, 2012] B Aditya Prakash, Jilles Vreeken, and Christos Faloutsos. Spotting culprits in epidemics: How many and which ones? In *2012 IEEE 12th International Conference on Data Mining*, pages 11–20. IEEE, 2012.
- [Shah and Zaman, 2011] Devavrat Shah and Tauhid Zaman. Rumors in a network: Who’s the culprit? *IEEE Transactions on Information Theory*, 57(8):5163–5181, 2011.
- [Shah *et al.*, 2020] Chintan Shah, Nima Dehmamy, Nicola Perra, Matteo Chinazzi, Albert-László Barabási, Alessandro Vespignani, and Rose Yu. Finding patient zero: Learning contagion source with graph neural networks. *arXiv preprint arXiv:2006.11913*, 2020.
- [Srinivasan and Ribeiro, 2019] Balasubramaniam Srinivasan and Bruno Ribeiro. On the equivalence between positional node embeddings and structural graph representations. In *International Conference on Learning Representations*, 2019.
- [Wang *et al.*, 2017] Zheng Wang, Chaokun Wang, Jisheng Pei, and Xiaojun Ye. Multiple source detection without knowing the underlying propagation model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.
- [Wang *et al.*, 2022] Junxiang Wang, Junji Jiang, and Liang Zhao. An invertible graph diffusion neural network for source localization. In *Proceedings of the ACM Web Conference 2022*, pages 1058–1069, 2022.
- [Wu *et al.*, 2015] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.
- [Yadati *et al.*, 2019] Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. Hypergen: A new method for training graph convolutional networks on hypergraphs. *Advances in Neural Information Processing Systems*, 32, 2019.
- [Yang *et al.*, 2020] Fan Yang, Shuhong Yang, Yong Peng, Yabing Yao, Zhiwen Wang, Houjun Li, Jingxian Liu, Ruisheng Zhang, and Chungui Li. Locating the propagation source in complex networks with a direction-induced search based gaussian estimator. *Knowledge-Based Systems*, 195:105674, 2020.
- [Yin *et al.*, 2022] Nan Yin, Fuli Feng, Zhigang Luo, Xiang Zhang, Wenjie Wang, Xiao Luo, Chong Chen, and Xian-Sheng Hua. Dynamic hypergraph convolutional network. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 1621–1634. IEEE, 2022.
- [Zhu *et al.*, 2017] Kai Zhu, Zhen Chen, and Lei Ying. Catch’em all: Locating multiple diffusion sources in networks with partial observations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.
- [Zhu *et al.*, 2022] Peican Zhu, Le Cheng, Chao Gao, Zhen Wang, and Xuelong Li. Locating multi-sources in social networks with a low infection rate. *IEEE Transactions on Network Science and Engineering*, 9(3):1853–1865, 2022.