DYNAMIC HYPERGRAPH SOURCE IDENTIFICATION VIA TOPOLOGY-AWARE MAMBA AND DIFFUSION SUPERVISION

Anonymous authors
Paper under double-blind review

ABSTRACT

Source identification in dynamic hypergraphs is challenging because evolving hyperedges continuously alter the available propagation paths, making diffusion dynamics strongly dependent on structural evolution. Existing methods mainly learn diffusion patterns from snapshot sequences, but often overlook dynamic higher-order interactions. Therefore, we propose a framework for dynamic hypergraph source identification via Topology-guided Mamba and Diffusion Supervision (TMDS). To capture the hyperedge evolution, the hyperedge evolution modeling module first characterizes node-level structural evolution from participation scale, neighbor composition, and group organization. These transition signals are then incorporated into a dynamic topology-aware Mamba, where retention and propagation gates adapt temporal inference to evolving higher-order structures. Furthermore, a forward diffusion supervision mechanism reconstructs the diffusion trajectory from the predicted source distribution and feeds the reconstruction discrepancy back to the inference model. Experiments on four diverse datasets demonstrate consistent improvements over representative baselines and verify the effectiveness of the proposed approach.

1 INTRODUCTION

Social networks facilitate large-scale information diffusion, but also accelerate the spread of harmful rumors (Notarmuzi et al., 2022), threatening public trust and social stability (Shelke & Attar, 2019). Source identification aims to identify the spreader of observed information cascades and has emerged as a critical issue in network science (Jiang et al., 2024). With increasingly complex network structures, hypergraph-based source identification has gained increasing attention for modeling higher-order relations (He et al., 2025; Cheng et al., 2025b).

Currently, hypergraph-based source identification methods typically employ temporal models to learn diffusion dynamics from observed snapshots and infer the most likely source nodes (Cheng et al., 2025c;b). However, these methods largely focus on static networks, neglecting the structural evolution of dynamic hypergraphs. In real-world social scenarios, higher-order interactions may evolve continuously as hyperedges form, disappear, or reorganize over time. As illustrated in Fig. 1, the teamwork group disappears after teamwork done, whereas the dormitory group changes when roommate moves out. During information diffusion, such evolution continually alters the available propagation paths and affects the diffusion process. Therefore, independently encoding each snapshot does not model structural dynamics over time, which may reduce source identification accuracy in dynamic hypergraphs. Moreover, current methods largely ignore whether the predicted source can explain the observed diffusion process. Under evolving topology, this issue is more pronounced as changing hyperedges continually reshape the available propagation paths.

The challenges of source identification in dynamic hypergraphs are summarized as follows: **1) Lack of topology evolution modeling.** Previous studies capture the network structure by encoding each snapshot separately. However, these methods fail to model the evolving higher-order interactions across snapshots, limiting their ability for dynamic hypergraphs. **2) Limited topology sensitivity in temporal models.** Existing temporal models capture sequential dependencies across snapshots. However, the learned propagation dynamics are insufficiently responsive to structural

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

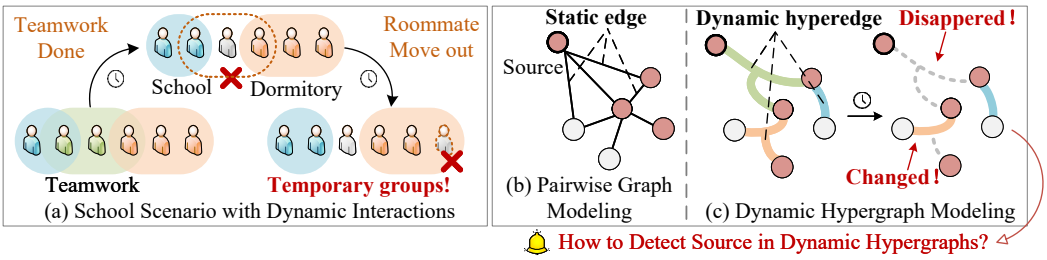


Figure 1: Social scenario modeling. (a) School scenario with dynamic interactions. (b) Pairwise graph modeling in current methods. (c) Dynamic hypergraph modeling in the proposed method, which captures the dynamic higher-order interactions for accurate source identification.

changes, which limits accurate source inference when propagation paths change over time. **3) The propagation reconstruction gap.** Current methods rely mainly on source-label supervision, without explicitly evaluating whether the resulting diffusion can reproduce the observed propagation process. This issue is particularly challenging for source identification in dynamic hypergraphs.

To address these limitations, we propose a framework for dynamic hypergraph source identification via Topology-guided Mamba and Diffusion Supervision (TMDS). To capture the hyperedge evolution, we quantifies node-level topology changes from three progressively finer dimensions, including participation scale, neighbor composition, and group organization. Subsequently, we develop the dynamic topology-aware Mamba module, which generates retention and propagation gates to adaptively control historical memory and higher-order information propagation. Moreover, to provide trajectory-level supervision, we further introduce a forward diffusion supervision mechanism that reconstructs the propagation trajectory from the predicted source distribution over the dynamic hypergraph sequence. The discrepancy between the reconstructed and observed sequences is fed back to the model, thereby encouraging the prediction to better account for the observed propagation.

Our main contributions are summarized as follows:

- To capture structural evolution in dynamic hypergraphs, we design a hyperedge evolution modeling module that quantifies node-level topological changes from three progressively finer dimensions, providing explicit structural context across snapshots.
- For enhancing awareness of topology dynamics in the inference process, we introduce a dynamic topology-aware Mamba that derives two topology-guided gates, enabling temporal inference to adapt historical information and current messages under structural evolution.
- Account for the propagation reconstruction gap, we introduce a forward diffusion supervision mechanism that propagates the predicted source distribution over the dynamic hypergraph, providing the trajectory-level supervision.

2 Related Work

Source Identification in Pairwise Graphs Source identification infers diffusion origins from partial propagation evidence (Ying & Zhu, 2022). Classical estimators exploit infection times (Pinto et al., 2012), diffusion directions (Yang et al., 2020), or label propagation for multiple sources (Wang et al., 2017). Meanwhile, observer-based methods identify sources from states or timestamps recorded at selected sensor nodes (Gajewski et al., 2022), while later work addresses structurally incomplete networks (Cheng et al., 2025a) and observer selection (Hu et al., 2024). Furthermore, learning-based methods formulate identification as node classification or inverse diffusion. Graph-convolutional formulations combine graph snapshots with propagation features (Dong et al., 2019; Cheng et al., 2024a), whereas later approaches incorporate attentive fusion (Cheng et al., 2024b), contrastive learning (Bao et al., 2024), and invertible diffusion inference (Wang et al., 2022; Ling et al., 2022). Recent work further targets scalability (Wang et al., 2023), incomplete observations (Kang et al., 2026), and cross-network (Ling et al., 2024). However, they typically operate on pairwise graphs, obscuring the higher-order pathways that govern diffusion.

Source Identification in Hypergraphs Hypergraph methods preserve group interactions explicitly (Benson et al., 2016; Antelmi et al., 2023), whose diffusion dynamics can depart substantially from those of pairwise networks (Majhi et al., 2022). General representation models encode fixed hypergraphs through spectral convolution (Zhou et al., 2006; Feng et al., 2019), convolution and attention (Bai et al., 2021), sparse graph approximations (Yadati et al., 2019), or learnable multiset functions (Chien et al., 2022). Huang & Yang (2021) unify graph and hypergraph message passing, while Cai et al. (2022) infer task-dependent hyperedges from data. DHGNN further reconstructs latent hyperedges from embeddings between network layers (Jiang et al., 2019); however, this layer-wise structure adaptation differs from modeling a sequence of externally observed topology changes. For source identification, HHAN models heterogeneous infection relations with hypergraph attention (He et al., 2025), and HyperDet fuses interactive and propagation-aware features (Cheng et al., 2025c). Despite their effectiveness, these methods largely encode individual observations without explicitly coupling reverse inference to the evolution of higher-order interactions.

Dynamic Graph and Hypergraph Learning Empirical studies of timestamped higher-order interactions show that group structure exhibits temporal organization that is lost under static or pairwise representations (Benson et al., 2018). Dynamic graph models capture related temporal variation by evolving GCN parameters across snapshots (Pareja et al., 2020), recurrently updating hierarchical node states (You et al., 2022), or encoding long interaction histories through co-occurrence features and temporal patches (Yu et al., 2023). These approaches establish effective temporal representation learning for pairwise graphs, but their dyadic topology and predominantly forward-prediction objectives do not directly address inverse diffusion over evolving hyperedges. TMDS instead treats the observed hypergraph sequence as part of the source-identification problem and uses explicit topology-transition descriptors to regulate reverse temporal inference.

State Space Models State space models (SSMs) encode sequences through recurrent latent dynamics and can admit parallel computation after discretization (Gu et al., 2020; 2021b). Structured SSMs retain long-range information with linear sequence complexity (Gu et al., 2021a), while Mamba makes their parameters input dependent to selectively retain or discard evidence (Gu & Dao, 2023). Graph extensions adapt Mamba to relational data through topology-aware tokenization (Behrouz & Hashemi, 2024), spatial state-space modeling (Choi et al., 2024), and learned dynamic graph structures (Yuan et al., 2025). GraphSSM develops state-space dynamics for temporal pairwise graphs by incorporating structural regularization into the continuous-time formulation (Li et al., 2024). More recently, Cheng et al. (2025b) propose SourceDetMamba for source identification, which applies Mamba to reverse-ordered hypergraph snapshots and augments each update with neighborhood information. However, GraphSSM targets general temporal-graph representation, whereas SourceDetMamba does not explicitly condition its state transitions on measured changes between consecutive hypergraph topologies. TMDS addresses this gap with topology-guided gates that adaptively balance historical information and current messages during reverse inference.

3 PRELIMINARIES

Dynamic Hypergraph Structures Let $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}^t)$ represent the hypergraph at time t, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ and $\mathcal{E}^t = \{e_1^t, \ldots, e_{M_t}^t\}$ denote the node set and hyperedge set, respectively. Each hyperedge $e \in \mathcal{E}^t$ contains a subset of nodes from $\mathcal{V}$. The structural relationships are represented by the incidence matrix $\mathbf{H}_t \in \{0,1\}^{N \times M_t}$, where:

$$\mathbf{H}_t^{ij} = \begin{cases} 1, & v_i \in e_j^t, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Typically, a dynamic hypergraph is represented as $\mathcal{G}_{1:T} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_T\}$, in which the hyperedge set $\mathcal{E}^t$ evolves over time to characterize time-varying higher-order interactions among nodes.

Snapshot-Based Source Localization Given a sequence of observed snapshots $O_{1:T}$ over a network $\mathcal{G}$, conventional snapshot-based methods encode each snapshot separately and process the resulting representations with a temporal model to estimate the source distribution:

$$\mathbf{Z}_t = \phi\left(\mathcal{G}, O_t\right), \qquad \mathbf{p} = f_\theta(\mathbf{Z}_{1:T}), \tag{2}$$

where ϕ and f_θ denote the snapshot encoder and the inference model, respectively, $\mathbf{Z}_t$ is the representation of snapshot t, and $\mathbf{p} = [p_1, \ldots, p_N]^\top$ contains the source probabilities of all nodes. However, this formulation captures structural information within each snapshot, neglecting the dynamic nature of the network in the inference process.

TMDS Framework Given propagation snapshots $O_{1:T}$ over the dynamic hypergraph $\mathcal{G}_{1:T}$, TMDS explicitly characterizes the hyperedge evolution signals $q_{1:T}$ between adjacent topologies, which guide the reverse inference model for source inference:

$$\mathbf{Z}_t = \phi(\mathcal{G}_t, O_t), \qquad \mathbf{p} = \mathcal{F}_\Theta\left(\mathbf{Z}_{T:1}, q_{T:1}\right), \tag{3}$$

where $q_t = \Delta(\mathcal{G}_{t-1}, \mathcal{G}_t)$ denotes the transition signal and $\mathcal{F}_\Theta$ denotes the Mamba model for inverse temporal inference. Based on the inferred source, TMDS reconstructs the diffusion process and compares the reconstructed cascade $\widehat{O}_{1:T}$ with the observed cascade $O_{1:T}$ to provide trajectory-level supervision.

$$\widehat{O}_{1:T} = \Phi\left(\mathbf{p}; \mathcal{G}_{1:T}\right) \tag{4}$$

Discussion TMDS is built on a dynamic view of source identification in hypergraphs. Unlike conventional snapshot-based approaches, it explicitly accounts for structural evolution during diffusion, allowing source inference to adapt to changing propagation pathways. Moreover, rather than relying solely on source-label supervision, TMDS evaluates whether the inferred sources can reproduce the observed cascade over the evolving topology, thereby introducing trajectory-level supervision.

4 METHOD

In this section, we present the TMDS framework for source identification in dynamic hypergraphs, which consists of three main components: Hyperedge Evolution Modeling, Dynamic Topology-Aware Mamba, and Forward Diffusion Supervision.

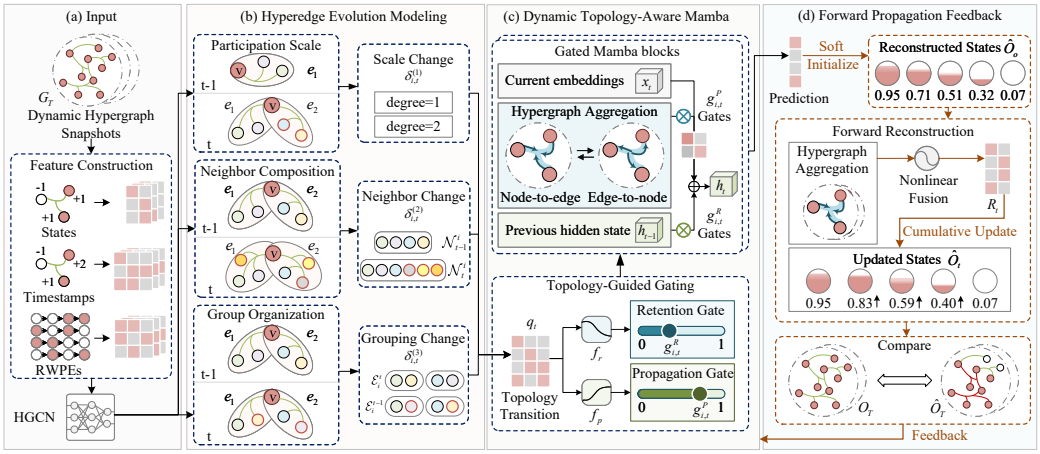


Figure 2: Overview of TMDS. (a) Input: infection states and structural positions are encoded by HGCNs. (b) Hyperedge Evolution Modeling: scale, neighbor, and grouping changes provide node-level transition signals. (c) Dynamic Topology-Aware Mamba: topology-guided gates adapt Mamba state updates to evolving structures. (d) Forward Diffusion Supervision: source predictions softly initialize forward reconstruction for trajectory supervision.

4.1 SNAPSHOT EMBEDDING

We first construct the initial node features by integrating propagation observations with structural position information. Specifically, we concatenate the observed propagation attributes with random-walk positional encodings (RWPEs):

$$\widetilde{\mathbf{X}}_t = [\mathbf{X}_t \,\|\, \mathbf{R}_t], \tag{5}$$

where $\mathbf{X}_t$ contains the node infection states and observed propagation timestamps, and $\mathbf{R}_t$ is the corresponding RWPE matrix.

Further, the hypergraph convolutional network (HGCN) encodes higher-order structural dependencies by propagating the combined features over the incidence matrix H_t:

$$\mathbf{E}_t^{(l+1)} = \mathrm{ReLU}\left(D_{v,t}^{-\frac{1}{2}} H_t D_{e,t}^{-1} H_t^\top D_{v,t}^{-\frac{1}{2}} \mathbf{E}_t^{(l)} \mathbf{W}^{(l)}\right), \tag{6}$$

where $\mathbf{E}_t^{(0)} = \widetilde{\mathbf{X}}_t$, $D_{v,t}$ and $D_{e,t}$ are the node degree and hyperedge degree matrices, respectively, $\mathbf{W}^{(l)}$ is the learnable parameter matrix. After L layers, we obtain $\mathbf{Z}_t = \mathbf{E}_t^{(L)}$, which captures the propagation state and higher-order structure within snapshot t.

4.2 Hyperedge Evolution Modeling

To capture structural evolution in dynamic hypergraphs, we introduce a hyperedge evolution modeling module, which quantifies node-level variations in local higher-order structure across adjacent snapshots. Specifically, this module characterizes this evolution from three progressively finer dimensions, including participation scale, neighbor composition, and group organization.

Participation Scale Participation scale quantifies changes in the number of hyperedges incident to node v_i through the normalized absolute difference

$$\delta_{i,t}^{(1)} = \frac{|d_{i,t} - d_{i,t-1}|}{d_{i,t} + d_{i,t-1} + \epsilon}, \tag{7}$$

where $d_{i,t}$ denotes the number of hyperedges incident to v_i at time t, and $\epsilon > 0$ prevents division by zero. A larger $\delta_{i,t}^{(1)}$ indicates a stronger change in the participation scale of v_i.

Neighbor Composition As hyperdegree alone cannot capture changes in interaction partners, we further characterize neighbor composition by comparing the neighbor sets of v_i across adjacent snapshots, formulated as:

$$\mathcal{N}_i^t = \left\{ v_j \,\middle|\, \exists e \in \mathcal{E}^t,\ v_i, v_j \in e,\ j \neq i \right\}, \tag{8}$$

where $\mathcal{E}^t$ denotes all hyperedges and $\mathcal{N}_i^t$ is the neighbor set of v_i at time t. The neighbor composition variation $\delta_{i,t}^{(2)}$ is the Jaccard distance between $\mathcal{N}_i^t$ and $\mathcal{N}_i^{t-1}$, with $\delta_{i,t}^{(2)} = 0$ when both sets are empty.

Group Organization To capture changes in group organization while the neighbor set remains unchanged, we characterize how the neighbors of v_i are repartitioned among hyperedges. Let $\mathcal{E}_i^\tau = \{e \in \mathcal{E}^\tau \mid v_i \in e\}$ denote the hyperedges incident to v_i at snapshot $\tau \in \{t-1, t\}$. We compare hyperedges across the two snapshots by measuring the Jaccard similarity of their members after excluding v_i:

$$C_{kl} = |(e_k \setminus \{v_i\}) \cap (e'_l \setminus \{v_i\})| \,/\, |(e_k \setminus \{v_i\}) \cup (e'_l \setminus \{v_i\})|. \tag{9}$$

To account for hyperedge splitting and merging, each hyperedge is matched to its most similar hyperedge in the adjacent snapshot. The best-match similarities in both temporal directions are

$$P_{i,t}^{(1)} = d_{i,t}^{-1} \sum\nolimits_l \max_k C_{kl} \qquad P_{i,t}^{(2)} = d_{i,t-1}^{-1} \sum\nolimits_k \max_l C_{kl}, \tag{10}$$

where $d_{i,\tau}$ denotes the hyperdegree of v_i at snapshot τ. The bidirectional comparison captures both hyperedge splits and merges. Accordingly, the group organization variation is:

$$\delta_{i,t}^{(3)} = 1 - \frac{1}{2}\left(P_{i,t}^{(1)} + P_{i,t}^{(2)}\right). \tag{11}$$

Consequently, the node-level hyperedge evolution is summarized as:

$$q_{i,t} = \left[\delta_{i,t}^{(1)}, \delta_{i,t}^{(2)}, \delta_{i,t}^{(3)}\right], \tag{12}$$

where $q_{i,t}$ is subsequently used by the topology-guided gates for state updating.

4.3 DYNAMIC TOPOLOGY-AWARE MAMBA

To learn the structural dynamics of the hypergraph, we propose a dynamic topology-aware Mamba module that performs reverse temporal inference over the snapshot representations through SSMs with topology-guided gating. Here, t indexes reverse processing from O_T to O_1, and $q_{i,t}$ describes the transition between the snapshots processed at steps $t-1$ and t, with $q_{i,1} = \mathbf{0}$. At each step, the module combines a state-space transition with hypergraph aggregation, while gates derived from $q_{i,t}$ control historical retention and hypergraph message injection.

Selective SSMs with Hypergraph Aggregation Typically, the structured State Space Model (SSM) evolves a latent state along the reverse-ordered sequence. Given x_t derived from $\mathbf{Z}_t$, it updates h_t and produces y_t as

$$h_t = \mathcal{A}h_{t-1} + \mathcal{B}x_t \qquad y_t = \mathcal{C}h_t, \tag{13}$$

where $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ denote the state transition, input, and output mappings, respectively.

To represent the higher-order relations, we incorporate hypergraph aggregation into SSMs. At each step, the historical node states are first aggregated into hyperedges to form hyperedge states. Subsequently, the hyperedge-level states E_t are propagated back to incident nodes, formulated as:

$$m_t = D_{v,t}^{-1} H_t E_t W_v \qquad E_t = \phi_e\big(D_{e,t}^{-1} H_t^\top h_{t-1} W_e\big), \tag{14}$$

where m_t therefore reflects the higher-order propagation message under snapshot t, $\phi_e(\cdot)$ is a nonlinear activation, and W_e and W_v are learnable parameter matrices.

Topology-Guided Gating To make the state update responsive to structural change, we introduce topology-guided gates that control historical retention and structural message injection according to $q_{i,t}$. Specifically, the retention and propagation gates are defined as

$$g_{i,t}^R = \exp\left(-w_r^\top q_{i,t}\right) \qquad g_{i,t}^P = 1 - \exp\left(-w_p^\top q_{i,t}\right). \tag{15}$$

We enforce nonnegativity through $w_r = \mathrm{softplus}(\theta_r)$ and $w_p = \mathrm{softplus}(\theta_p)$, where θ_r and θ_p are trainable vectors. Since each component of $q_{i,t}$ lies in $[0,1]$, this parameterization guarantees $g_{i,t}^R \in (0,1]$ and $g_{i,t}^P \in [0,1)$. Larger topology variations therefore decrease historical retention and increase the contribution of the current hypergraph message. The gated state-space update is

$$h_{i,t} = \mathcal{A}\left(g_{i,t}^R h_{i,t-1}\right) + \mathcal{B}x_{i,t} + \left(1 + g_{i,t}^P\right) m_{i,t}. \tag{16}$$

where $h_{i,t-1}$ is the previous hidden state and $x_{i,t}$ is the current snapshot embedding. Appendix C.3 reports the stability analysis of the learned gates to topology variation. Based on the final representations, the source probability of each node is estimated as

$$\mathbf{p} = \mathrm{sigmoid}\left(f_s(h_{out})\right). \tag{17}$$

where f_s denotes the MLP head and $\mathbf{p} = [p_1, \ldots, p_N]^\top$ contains the predicted source probabilities. The sigmoid is applied element-wise, allowing multiple nodes to receive high source probabilities independently rather than forcing the scores to sum to one.

4.4 FORWARD DIFFUSION SUPERVISION

Given the source probabilities, the forward diffusion supervision mechanism reconstructs the cascade for explaining the observed trajectory. Predicted source probabilities initialize the cascade as

$$\widehat{O}_{i,0} = p_i, \qquad i = 1, \ldots, N. \tag{18}$$

At snapshot t, normalized node-to-hyperedge and hyperedge-to-node propagation computes the exposure received by all nodes:

$$\mathbf{U}_t = \mathbf{D}_{v,t}^{-1} \mathbf{H}_t \mathbf{D}_{e,t}^{-1} \mathbf{H}_t^\top \widehat{\mathbf{O}}_{t-1}. \tag{19}$$

Here, $\mathbf{H}_t^\top$ aggregates node states into hyperedges, while $\mathbf{H}_t$ propagates the averaged hyperedge states back to incident nodes. Afterwards, we transform the exposure $\mathbf{U}_t$ into a response vector $\mathbf{R}_t$ and update the reconstructed state:

$$\mathbf{R}_t = 1 - \exp\left(-s\mathbf{U}_t\right), \qquad \widehat{\mathbf{O}}_t = \widehat{\mathbf{O}}_{t-1} + \left(\mathbf{1} - \widehat{\mathbf{O}}_{t-1}\right) \odot \alpha \mathbf{R}_t. \tag{20}$$

We parameterize $\alpha = \mathrm{sigmoid}(\widetilde{\alpha})$ and $s = \mathrm{softplus}(\widetilde{s})$, which keeps the reconstructed states in $[0, 1]$ under the normalized exposure operator. The reconstructed snapshots are compared with the observations using balanced binary cross-entropy, yielding the discrepancy $\mathcal{D}_t$ and trajectory-level supervision. The discrepancy details are specified in Appendix A.2. In TMDS, FDS serves to regularize source predictions toward forward consistency within the evaluated regime, rather than to identify a general higher-order propagation mechanism.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate TMDS on House (Chodrow et al., 2021), Cora (McCallum et al., 2000), NTU (Chen et al., 2003), and Test-Bills (Benson et al., 2018). House, Cora, and NTU contain 1,290/341, 1,072/2,388, and 2,012/2,012 nodes/hyperedges, respectively.

Baselines. We compare TMDS with three graph-based methods, GIN-SD (Cheng et al., 2024b), IVGD (Wang et al., 2022), and SL-VAE (Ling et al., 2022), and three hypergraph-based methods, HHAN (He et al., 2025), HyperDet (Cheng et al., 2025c), and SDMamba (Cheng et al., 2025b). All methods use identical cascade splits, observations, candidate nodes, and source labels. Appendix B.3 specifies the details for every baseline.

Metrics. We report accuracy (ACC), class-weighted F1-score (F1), and area under the receiver operating characteristic curve (AUC), with higher values indicating better performance. Appendix B.1 provides their definitions.

Implementation. For real-world static hypergraphs, we first specify the topology evolution intensity and introduce random hyperedge additions, deletions, and reorganizations during information diffusion. Propagation is controlled by the propagation rate β and noise level, with sources initialized according to a fixed source-node ratio. The evolving network states are then sampled at a predefined snapshot interval to construct the observation sequence used for source identification. For Test-Bills, the six ordered snapshots directly provide the dynamic topology and are used without additional structural simulation. Appendix B.2 reports the configuration details.

5.2 Main Results

Comparison with State-of-the-Art Methods To evaluate overall source-identification performance, we compare TMDS with three representative graph-based baselines and three hypergraph-based baselines on House, Cora, NTU, and Test-Bills under two propagation rates $\beta \in \{0.15, 0.25\}$. Table 1 shows that TMDS consistently achieves the best performance across all datasets and metrics. At $\beta = 0.15$, it improves the strongest baseline by 4.9 F-score points on Cora and 5.0 AUC points on Test-Bills. These gains increase to 5.7 and 6.0 points at $\beta = 0.25$, respectively, showing that the advantage persists across the two evaluated propagation rates.

Visualization Results To visualize the results, we apply TMDS and three competitive hypergraph-based baselines to the same Test-Bills diffusion instance and compare their predicted sources against the ground truth. As shown in Figure 3, TMDS more accurately recovers ground-truth sources across both the core and branches than the three competitive baselines.

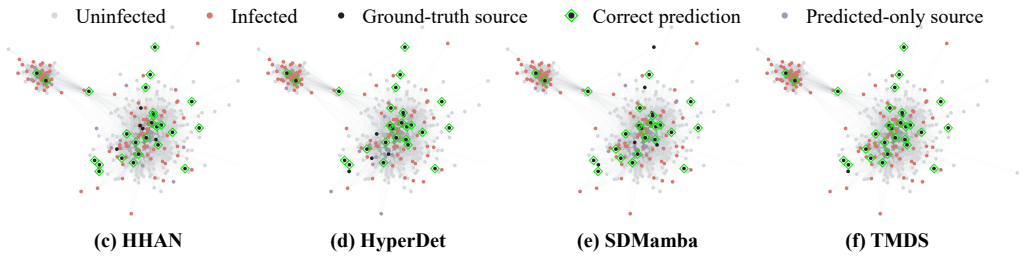


Figure 3: Visualization results on Test-Bills network.

Table 1: Performance comparison with state-of-the-art methods across four datasets. Best results are shown in bold, and second-best results are underlined.

β	Method	House			Cora			NTU			Test-Bills		
		ACC	F-score	AUC	ACC	F-score	AUC	ACC	F-score	AUC	ACC	F-score	AUC
0.15	GIN-SD	0.826	0.673	0.830	0.729	0.586	0.731	0.775	0.576	0.793	0.814	0.649	0.811
	IVGD	0.863	0.742	0.878	0.737	0.502	0.832	0.861	0.739	0.885	0.830	0.755	0.865
	SL-VAE	0.801	0.626	0.813	0.738	0.693	0.903	0.826	0.672	0.885	0.842	0.701	0.854
	HHAN	0.872	0.835	0.902	0.772	0.720	0.815	0.875	0.830	0.959	0.815	0.840	0.855
	HyperDet	0.801	0.849	0.800	0.852	0.842	0.739	0.911	0.926	0.800	0.854	0.857	0.865
	SDMamba	0.886	0.918	0.921	0.793	0.868	0.837	0.907	0.928	0.965	0.812	0.881	0.873
	TMDS	**0.905**	**0.940**	**0.929**	**0.869**	**0.917**	**0.919**	**0.919**	**0.949**	**0.967**	**0.859**	**0.916**	**0.923**
0.25	GIN-SD	0.768	0.621	0.792	0.683	0.552	0.706	0.724	0.531	0.756	0.773	0.612	0.782
	IVGD	0.811	0.701	0.846	0.704	0.481	0.806	0.806	0.698	0.851	0.826	0.718	0.839
	SL-VAE	0.762	0.601	0.791	0.711	0.668	0.879	0.783	0.641	0.858	0.809	0.674	0.831
	HHAN	0.824	0.798	0.879	0.735	0.691	0.794	0.831	0.795	0.931	0.787	0.803	0.829
	HyperDet	0.779	0.821	0.786	0.816	0.809	0.728	0.834	0.881	0.783	0.821	0.825	0.842
	SDMamba	0.839	0.874	0.899	0.758	0.831	0.816	0.831	0.884	0.942	0.784	0.843	0.851
	TMDS	**0.842**	**0.883**	**0.907**	**0.824**	**0.888**	**0.901**	**0.835**	**0.893**	**0.944**	**0.835**	**0.896**	**0.911**

Impact of Topology Evolution Intensity Furthermore, we compare TMDS with the representative baseline SDMamba under different topology evolution intensities $\rho \in \{5\%, 15\%, 25\%, 35\%\}$. For Test-Bills, this experiment applies the topology perturbations to its first snapshot as the initial hypergraph. As shown in Figure 4, TMDS consistently achieves higher F-scores, with gains generally increasing as topology evolution intensifies. In particular, the improvement grows from 2.0 to 6.8 points on Cora and from 1.0 to 6.4 points on Test-Bills, demonstrating the benefit of explicitly modeling hyperedge transitions under stronger structural evolution.

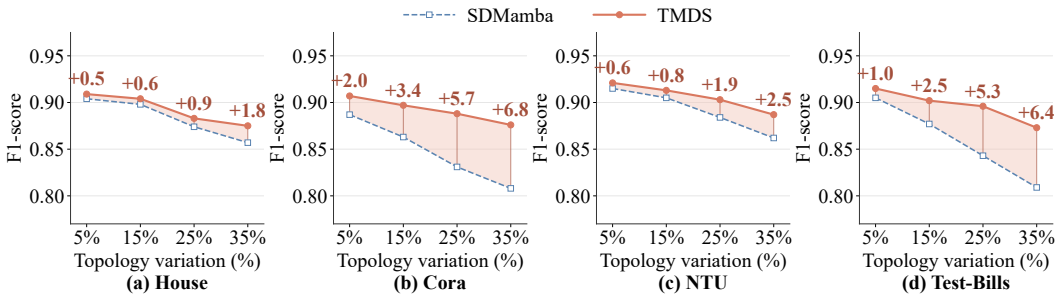
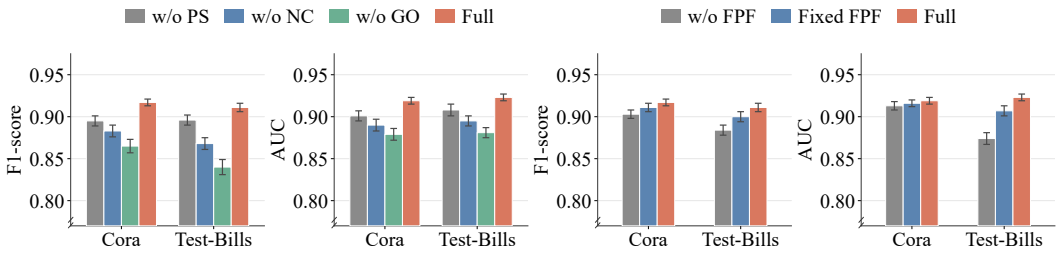


Figure 4: Performance comparison under increasing topology-evolution intensity.

5.3 Ablation Study and Analysis

Ablation of Topology-Guided Gating To evaluate the topology-guided gating design, we consider six variants: 1) no topology conditioning, with $G^R = 1$ and $G^P = 0$; 2) shared global gates across all nodes and snapshots; 3) feature-conditioned gates generated from snapshot embeddings instead of $q_{i,t}$; 4) shuffled transition descriptors, where the node–time alignment of $q_{i,t}$ is disrupted while preserving its distribution; 5) w/o G^P, which removes adaptive propagation enhancement; and 6) w/o G^R, which removes adaptive historical decay. The full gating design consistently performs best as shown in Table 2. Global, feature-conditioned, or shuffled alternatives all degrade performance, confirming the need for correctly aligned topology-transition signals. Removing either G^P or G^R also reduces performance, validating their complementary roles. Alternative temporal backbones are compared in Appendix D.1.



(a) Ablation of Topology Transition Modeling (TTM) **(b) Ablation of Forward Propagation Feedback (FPF)**

Figure 5: Ablation results on Cora and Test-Bills. Error bars denote the standard deviation.

Table 2: Ablation results of topology-guided gating.

Variant	Cora		Test-Bills	
	F1	AUC	F1	AUC
No topology conditioning	0.884 ± 0.006	0.878 ± 0.007	0.882 ± 0.007	0.869 ± 0.008
Fixed global gates	0.890 ± 0.006	0.885 ± 0.006	0.887 ± 0.007	0.878 ± 0.007
Feature-conditioned gates	0.899 ± 0.005	0.895 ± 0.005	0.895 ± 0.006	0.889 ± 0.006
Shuffled transition descriptors	0.886 ± 0.007	0.880 ± 0.007	0.884 ± 0.008	0.875 ± 0.008
w/o G^P	0.903 ± 0.005	0.895 ± 0.006	0.879 ± 0.007	0.872 ± 0.008
w/o G^R	0.906 ± 0.004	0.897 ± 0.005	0.888 ± 0.006	0.883 ± 0.007
Full	$\mathbf{0.917 \pm 0.004}$	$\mathbf{0.919 \pm 0.004}$	$\mathbf{0.911 \pm 0.005}$	$\mathbf{0.923 \pm 0.004}$

Ablation of Hyperedge Evolution Modeling To evaluate the contribution of components in HEM module, we remove Participation Scale (PS), Neighbor Composition (NC), and Group Organization (GO) individually under Cora and Test-Bills, leaving the topology-guided gates unchanged. As Figure 5(a) shows, removing any of the three components leads to performance degradation, while the full model consistently achieves the best results. This confirms that PS, NC, and GO all contribute to effective hyperedge evolution modeling. Particularly, removing GO causes the largest drop, indicating that changes in hyperedge organization provide important structural cues.

Ablation of Forward Diffusion Supervision To examine the contribution of Forward Diffusion Supervision (FDS), we evaluate three configurations while keeping the remaining model unchanged: 1) removing FDS module; 2) fixing the propagation parameters α and s to empirical values; and 3) learning both parameters from the same initialization. As shown in Figure 5(b), FDS consistently improves performance, confirming the benefit of forward-trajectory supervision. Learning α and s further yields the best results on both datasets, showing the advantage of adapting the reconstruction dynamics. The 2×2 interaction between topology gates and FDS is reported in Appendix D.2.

Complexity Analysis For fixed representation dimensions, the trainable components of TMDS scale linearly with the number of snapshots and sparse node–hyperedge incidences. Exact group-organization matching is computed once and cached, but remains quadratic in local hyperdegree; Appendix C.1 provides the complete analysis and runtime diagnostic.

6 Conclusion

We addressed source identification in dynamic hypergraphs by coupling explicit topology-transition modeling, topology-guided selective updates, and forward diffusion supervision. TMDS ranks first in all comparisons, while ablations and sensitivity tests support the contributions of its components and its stability within the tested ranges. The evidence is limited to SI-like cascades and future work will examine recovery and nonlinear group synergy.

AI USE STATEMENT

In this work, generative AI tools were used to aid or polish writing and to support literature retrieval and discovery. Their use was limited to these purposes. All AI-assisted text was reviewed by the authors, and all retrieved references were checked against the corresponding sources. The authors take full responsibility for the final manuscript, including its claims, citations, analysis, and presentation.

ETHICS STATEMENT

This work follows the ICLR Code of Ethics. Experiments use existing benchmark hypergraphs and simulated noisy SI cascades; no new human participants or personally identifiable data are collected. Although source identification supports diffusion analysis, it may enable surveillance or unsupported attribution. Predictions should not be used as sole evidence for identifying or penalizing individuals. Deployment should follow applicable data-governance requirements and independent domain review. The authors take responsibility for the reported methods and results.

REPRODUCIBILITY STATEMENT

The task formulation is defined in Section 3, while Section 4 specifies the architecture, transition descriptors, gated updates, and forward feedback. Appendix A gives the remaining formulations; Appendix B reports cascade counts, data splits, optimization, training schedules, representation dimensions, loss settings, and baseline adaptations; and Appendix C reports scalability and gate-stability diagnostics. Section 5 and Appendix E provide the main comparisons, ablations, and sensitivity analyses. Code will be released upon acceptance.

REFERENCES

Alessia Antelmi, Gennaro Cordasco, Mirko Polato, Vittorio Scarano, Carmine Spagnuolo, and Dingqi Yang. A survey on hypergraph representation learning. *ACM Computing Surveys*, 56 (1):1–38, 2023.

Song Bai, Feihu Zhang, and Philip H. S. Torr. Hypergraph convolution and hypergraph attention. *Pattern Recognition*, 110(107637), 2021.

Qing Bao, Ying Jiang, Wang Zhang, Pengfei Jiao, and Jing Su. Graph contrastive learning for source localization in social networks. *Information Sciences*, 679:121090, 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*, pp. 119–130, 2024.

Austin R Benson, David F Gleich, and Jure Leskovec. Higher-order organization of complex networks. *Science*, 353(6295):163–166, 2016.

Austin R. Benson, Rediet Abebe, Michael T. Schaub, Ali Jadbabaie, and Jon Kleinberg. Simplicial closure and higher-order link prediction. *Proceedings of the National Academy of Sciences*, 2018. ISSN 0027-8424. doi: 10.1073/pnas.1800683115.

Derun Cai, Moxian Song, Chenxi Sun, Baofeng Zhang, Shenda Hong, and Hongyan Li. Hypergraph structure learning for hypergraph neural networks. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence*, pp. 1923–1929, 2022.

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, pp. 223–232. Wiley Online Library, 2003.

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*, 54(11):7002–7014, 2024a.

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*, pp. 55–63, 2024b.

Le Cheng, Peican Zhu, Chao Gao, Zhen Wang, and Xuelong Li. Sdsi: Source detection in structurally incomplete social networks. *IEEE Transactions on Network Science and Engineering*, 12 (2):1041–1052, 2025a.

Le Cheng, Peican Zhu, Yangming Guo, Chao Gao, Zhen Wang, and Keke Tang. Sourcedetmamba: A graph-aware state space model for source detection in sequential hypergraphs. In *Proceedings of the 34th International Joint Conference on Artificial Intelligence*, pp. 2749–2757, 2025b.

Le Cheng, Peican Zhu, Yangming Guo, Keke Tang, Chao Gao, and Zhen Wang. Hyperdet: source detection in hypergraphs via interactive relationship construction and feature-rich attention fusion. In *Proceedings of the 34th International Joint Conference on Artificial Intelligence*, pp. 2758–2766, 2025c. ISBN 978-1-956792-06-5.

Eli Chien, Chao Pan, Jianhao Peng, and Olgica Milenkovic. You are allset: A multiset function framework for hypergraph neural networks. In *International Conference on Learning Representations*, 2022.

Kyunghyun Cho, Bart Van Merriënboer, Çağlar Gulçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1724–1734, 2014.

Philip S Chodrow, Nate Veldt, and Austin R Benson. Generative hypergraph clustering: From blockmodels to modularity. *Science Advances*, 7(28):eabh1303, 2021.

Jinhyeok Choi, Heehyeon Kim, Minhyeong An, and Joyce Jiyoung Whang. Spot-mamba: Learning long-range dependency on spatio-temporal graphs with selective state spaces. *arXiv preprint arXiv:2406.11244*, 2024.

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*, pp. 569–578, 2019.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI conference on Artificial Intelligence*, pp. 3558–3565, 2019.

Łukasz G Gajewski, Robert Paluch, Krzysztof Suchecki, Adam Sulik, Boleslaw K Szymanski, and Janusz A Hołyst. Comparison of observer based methods for source localisation in complex networks. *Scientific Reports*, 12(1):5079, 2022.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

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.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021a.

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, 2021b.

Qiang He, Yunting Bao, Hui Fang, Yuting Lin, and Hao Sun. Hhan: Comprehensive infectious disease source tracing via heterogeneous hypergraph neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 291–299, 2025.

Zhao-Long Hu, Hong-Jue Wang, Lei Sun, Chang-Bing Tang, and Minglu Li. Source localization in complex networks with optimal observers based on maximum entropy sampling. *Expert Systems with Applications*, 256:124946, 2024. ISSN 0957-4174.

Jing Huang and Jie Yang. Unignn: a unified framework for graph and hypergraph neural networks. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, pp. 2563–2569, 2021.

Jianwen Jiang, Yuxuan Wei, Yifan Feng, Jingxuan Cao, and Yue Gao. Dynamic hypergraph neural networks. In *Proceedings of International Joint Conferences on Artificial Intelligence*, pp. 2635–2641, 2019.

Yinan Jiang, Ranran Wang, Jianshan Sun, Yashen Wang, Haofang You, and Yin Zhang. Rumor localization, detection and prediction in social network. *IEEE Transactions on Computational Social Systems*, 11(3):3168–3178, 2024.

Yan Kang, Le Cheng, Peican Zhu, Zun Liu, Jiang Long, and Yangming Guo. Enhanced source detection in incomplete networks via redundancy-guided sensing and anchor-bounded inference. *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1–13, 2026.

Jintang Li, Ruofan Wu, Xinzhou Jin, Boqun Ma, Liang Chen, and Zibin Zheng. State space models on temporal graphs: A first-principles study. *Advances in Neural Information Processing Systems*, 37:127030–127058, 2024.

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*, pp. 1010–1020, 2022.

Chen Ling, Tanmoy Chowdhury, Jie Ji, Sirui Li, Andreas Züfle, and Liang Zhao. Source localization for cross network information diffusion. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5419–5429, 2024.

Soumen Majhi, Matjaž Perc, and Dibakar Ghosh. Dynamics on higher-order networks: A review. *Journal of the Royal Society Interface*, 19(188):43–58, 2022.

Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.

Daniele Notarmuzi, Claudio Castellano, Alessandro Flammini, Dario Mazzilli, and Filippo Radicchi. Universality, criticality and complexity of information propagation in social media. *Nature communications*, 13(1):1308, 2022.

Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao B. Schardl, and Charles E. Leiserson. EvolveGCN: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

Pedro C Pinto, Patrick Thiran, and Martin Vetterli. Locating the source of diffusion in large-scale networks. *Physical Review Letters*, 109(6):068702, 2012.

Sushila Shelke and Vahida Attar. Source detection of rumor in social network–a review. *Online Social Networks and Media*, 9:30–42, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Junxiang Wang, Junji Jiang, and Liang Zhao. An invertible graph diffusion neural network for source localization. In *Proceedings of the ACM Web Conference 2022*, pp. 1058–1069, 2022.

Zhen Wang, Dongpeng Hou, Chao Gao, Xiaoyu Li, and Xuelong Li. Lightweight source localization for large-scale social networks. In *Proceedings of the ACM Web Conference 2023*, pp. 286–294, 2023.

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*, pp. 279–308, 2017.

Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. Hypergcn: A new method for training graph convolutional networks on hypergraphs. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

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.

Lei Ying and Kai Zhu. *Diffusion source localization in large networks*. Springer Nature, 2022.

Jiaxuan You, Tianyu Du, and Jure Leskovec. Roland: Graph learning framework for dynamic graphs. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 2358–2366, 2022.

Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. Towards better dynamic graph learning: New architecture and unified library. *Advances in Neural Information Processing Systems*, 36:67686–67700, 2023.

Haonan Yuan, Qingyun Sun, Zhaonan Wang, Xingcheng Fu, Cheng Ji, Yongjian Wang, Bo Jin, and Jianxin Li. Dg-mamba: Robust and efficient dynamic graph structure learning with selective state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 22272–22280, 2025.

Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in Neural Information Processing Systems*, pp. 378–382, 2006.

A Detailed Transition and Feedback Formulations

A.1 Group-Organization Descriptor

Equations 9 and 11 define the group-organization descriptor. We set $C_{kl} = 1$ if both noncentral-member sets in Eq. 9 are empty. Similarly, $\delta_{i,t}^{(3)} = 0$ if both incident-hyperedge collections are empty and $\delta_{i,t}^{(3)} = 1$ if only one is empty. In all other cases, the descriptor requires $d_{i,t-1}d_{i,t}$ local hyperedge-pair comparisons. Let $c_J(e_k, e'_l)$ denote the cost of intersecting and uniting the two noncentral-member sets. The total preprocessing time is

$$O\left(\sum_{t=2}^{T}\sum_{i=1}^{N}\sum_{e_k\in\mathcal{E}_i^{t-1}}\sum_{e'_l\in\mathcal{E}_i^{t}} c_J(e_k, e'_l)\right), \tag{A.1}$$

which reduces to $O(\sum_{t,i} d_{i,t-1}d_{i,t}\bar{k}_{i,t})$ when the compared member sets have average size $\bar{k}_{i,t}$. The implementation streams the row-wise and column-wise maxima instead of materializing the complete similarity matrix, then caches the descriptors for subsequent epochs. This reduces auxiliary storage, although the worst-case time remains quadratic in local hyperdegree. Appendix C.2 measures this dependence directly.

A.2 Forward Diffusion Supervision

Let $\overline{O}_{t,i} = \mathrm{clip}(\widehat{O}_{t,i}, \epsilon, 1-\epsilon)$. The discrepancy between the observed and reconstructed states is

$$\mathcal{D}_t = -\frac{1}{N}\sum_{i=1}^{N}\left[w_t^+ O_{t,i}\log\overline{O}_{t,i} + (1-O_{t,i})\log\left(1-\overline{O}_{t,i}\right)\right]. \tag{A.2}$$

To prevent inactive nodes from dominating this loss, we use a capped inverse-frequency weight. For $N_t^+ \geq 1$ observed positive nodes, it is defined as

$$w_t^+ = \min\left(\kappa, \max\left(1, \frac{N-N_t^+}{N_t^+}\right)\right), \tag{A.3}$$

where κ caps the contribution of highly imbalanced snapshots. Before evaluating the logarithms, reconstructed probabilities are clipped to $[\epsilon, 1-\epsilon]$ for numerical stability.

A.3 Loss Function

The objective jointly optimizes source identification and trajectory consistency. Let $\mathcal{S}$ denote the ground-truth source set. The source and trajectory losses are

$$\mathcal{L}_{\mathrm{source}} = -\sum_{i\in\mathcal{S}}\log p_i - \omega\sum_{i\in\mathcal{V}\setminus\mathcal{S}}\log\left(1-p_i\right), \qquad \mathcal{L}_{\mathrm{traj}} = \frac{1}{T}\sum_{t=1}^{T}\mathcal{D}_t, \tag{A.4}$$

where ω balances source and non-source candidates, and $\mathcal{L}_{\mathrm{traj}}$ averages the snapshot discrepancies. The overall training objective is

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{source}} + \lambda\mathcal{L}_{\mathrm{traj}}, \tag{A.5}$$

where $\lambda \geq 0$ controls the contribution of trajectory-consistency supervision.

B Reproducibility and Baseline Protocol

B.1 Evaluation Metrics

Let $y_i \in \{0,1\}$ denote whether candidate node $v_i \in \mathcal{V}$ is a ground-truth source, and let $\widehat{y}_i \in \{0,1\}$ denote its predicted label. For evaluation, the predicted source set consists of the $|\mathcal{S}|$ candidate nodes with the largest sigmoid scores, matching the number of ground-truth sources. Let TP, TN,

FP, and FN denote the resulting numbers of true positives, true negatives, false positives, and false negatives, respectively. Accuracy is computed as

$$\mathrm{ACC} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{B.1}$$

Because source nodes form the minority class, we report the class-weighted F-score rather than an unweighted binary F-score. The positive- and negative-class F-scores and their support-weighted average are

$$F_1^+ = \frac{2TP}{2TP + FP + FN}, \qquad F_1^- = \frac{2TN}{2TN + FP + FN}, \qquad F_{1,\mathrm{weighted}} = \frac{(TP + FN)F_1^+ + (TN + FP)F_1^-}{TP + TN + FP + FN}. \tag{B.2}$$

This weighting assigns each class a contribution proportional to its support and therefore accounts for the strong imbalance between source and non-source nodes. AUC is computed from the continuous sigmoid scores by varying the decision threshold and measuring the area under the receiver operating characteristic curve:

$$\mathrm{AUC} = \int_0^1 \mathrm{TPR}(u)\,\mathrm{d}u, \tag{B.3}$$

where u denotes the false-positive rate and $\mathrm{TPR}(u)$ is the corresponding true-positive rate.

B.2 Data Splits and Training Details

Observations generated from the same source realization remain in the same training, validation, or test split. Every method uses the same instances and split indices. We select models by validation F-score and average test results over the random seeds listed in Table 4. Table 3 reports the number of cascades and the observation length for each dataset.

Table 3: Cascade-level data splits and observation lengths.

Dataset	Train	Validation	Test	T
House	6,000	1,000	2,000	3
Cora	6,000	1,000	2,000	3
NTU	6,000	1,000	2,000	3
Test-Bills	3,000	500	1,000	6

TMDS is trained end to end with AdamW. We use early stopping to limit overfitting and apply the same stopping criterion across datasets. The source-class weight ω is computed from the source/non-source ratio of each training instance, while λ and κ are selected on the validation set. Unless stated otherwise, the settings in Table 4 are shared across datasets.

Table 4: Optimization, training schedule, and representation dimensions.

Setting	Value
Optimizer	AdamW
Initial learning rate / weight decay	10^{-3} / 5×10^{-4}
Learning-rate schedule	10-epoch warm-up, cosine decay to 10^{-5}
Batch size / maximum epochs	32 / 300
Early-stopping patience	30 epochs
Random seeds	$\{0, 1, 2, 3, 4\}$
HGCN layers / dropout	2 / 0.2
Node hidden dimension	64
RWPE dimension	16
Mamba state dimension	64
Trajectory weight λ / cap κ	0.1 / 20
Initial α / s	0.65 / 2.0
Software / accelerator	PyTorch 1.13.1 / NVIDIA A100

B.3 Baseline Adaptation and Fairness

All baselines receive the same observed infection states, normalized timestamps, candidate nodes, and source labels. Each hypergraph snapshot is converted into the same weighted 2-section graph for graph-based methods:

$$\widetilde{A}_t = H_t D_{e,t}^{-1} H_t^\top, \qquad A_t = \widetilde{A}_t - \mathrm{diag}(\widetilde{A}_t), \tag{B.4}$$

This conversion prevents large hyperedges from contributing disproportionate pairwise weight. Static methods process all observed snapshots independently with shared parameters, and their node-level logits are averaged before source ranking. The parameter-free fusion exposes each method to the full observation sequence without adding a temporal model to its original architecture. Sequential methods process the same snapshots directly in reverse chronological order. We retain architecture-specific positional encodings and exclude the topology-transition descriptor and FDS loss introduced by TMDS from all baselines.

For each baseline, we retain its original objective and search the learning rate, hidden width, dropout, and method-specific coefficients on the common validation split. Every method receives the same number of validation trials, uses the same stopping criterion, and is evaluated with the same seed set. The comparison therefore controls the observations and selection budget while preserving each baseline's computational mechanism.

C Scalability and Stability Diagnostics

C.1 Time Complexity Analysis

Let N be the number of nodes, M_t the number of hyperedges, $S_t = \mathrm{nnz}(H_t)$ the number of node–hyperedge incidences at snapshot t, d the hidden dimension, d_s the Mamba state dimension, and L the number of HGCN layers. The topology-transition descriptors and RWPEs are computed once and cached. Denoting the implementation-dependent RWPE construction cost by $\mathcal{V}_{\mathrm{RWPE}}$, the preprocessing complexity is dominated by exact group-organization matching:

$$\mathcal{V}_{\mathrm{pre}} = O\left(\mathcal{V}_{\mathrm{RWPE}} + \sum_{t=2}^{T}\sum_{i=1}^{N} d_{i,t-1} d_{i,t} \bar{k}_{i,t}\right), \tag{C.1}$$

where $\bar{k}_{i,t}$ is the average size of the compared hyperedge-member sets. Appendix C.2 empirically examines this one-time cost.

For each sequence, sparse HGCN encoding costs $O(L\sum_t (Nd^2 + S_t d))$. The topology-aware Mamba scan, including its dense projections and node–hyperedge–node messages, costs $O(TN(d^2 + dd_s) + \sum_t (M_t d^2 + S_t d))$. The scalar FDS reconstruction costs $O(\sum_t (S_t + N))$, and a fixed-depth, width-d source head costs $O(Nd^2)$. Absorbing these terms, the complete trainable forward pass has complexity

$$\mathcal{V}_{\mathrm{forward}} = O\left(L\sum_{t=1}^{T}(Nd^2 + S_t d) + TN(d^2 + dd_s) + \sum_{t=1}^{T}(M_t d^2 + S_t d)\right). \tag{C.2}$$

The backward pass has the same asymptotic order. For B training sequences and E epochs, training costs $O(B\mathcal{V}_{\mathrm{pre}} + EB\mathcal{V}_{\mathrm{forward}})$. With fixed representation dimensions, TMDS is linear in the number of snapshots and sparse incidences during training and inference. Its only quadratic dependence on local structural size is the cached factor $d_{i,t-1}d_{i,t}$ required by exact matching at high-degree nodes.

C.2 Runtime Scaling of the Group-Organization Descriptor

We isolate the cost of $\delta_{i,t}^{(3)}$ from neural-network execution by constructing two consecutive local snapshots around an anchor node with $d_{i,t-1} = d_{i,t} = D$. Each incident hyperedge contains a fixed number of noncentral members. We evaluate the exact bidirectional maximum matching in Eq. 10 on a single CPU thread, discard warm-up runs, and report the mean and standard deviation over repeated measurements. This setup measures the dependence on local degree without including dataset-loading or GPU-kernel costs.

Table 5: Baseline adaptation protocol. Method-specific hyperparameters are selected by validation F-score under the same search budget.

Method	Structure	Dynamic-topology input	Temporal handling	Node features and positional encoding
GIN-SD (Cheng et al., 2024b)	Graph	Weighted 2-section A_t	Shared snapshot encoder; mean-logit fusion	Infection state, timestamp, and per-snapshot Laplacian PE
IVGD (Wang et al., 2022)	Graph	Weighted 2-section A_t	Shared inverse model; mean-logit fusion	Infection state and timestamp; original validity constraints retained
SL-VAE (Ling et al., 2022)	Graph	Weighted 2-section A_t	Shared variational model; mean posterior-logit fusion	Infection state and timestamp; original latent prior and objective retained
HHAN (He et al., 2025)	Hypergraph	Native incidence matrix H_t	Shared snapshot encoder; mean-logit fusion	Infection state and timestamp; a single node and hyperedge type when labels are unavailable
HyperDet (Cheng et al., 2025c)	Hypergraph	Native incidence matrix H_t	Shared snapshot encoder; mean-logit fusion	Infection state, timestamp, and its original positional feature construction
SDMamba (Cheng et al., 2025b)	Sequential hypergraph	Native sequence $H_{1:T}$	Native reverse-order sequence model	Infection state, timestamp, and the original structural encoding
TMDS	Dynamic hypergraph	Native sequence $H_{1:T}$	Reverse-order topology-guided Mamba	Infection state, timestamp, RWPE, and topology-transition descriptors

Table 6: Runtime stress test for the exact group-organization descriptor. Hyperedge size is fixed at 10; runtimes are milliseconds per node-transition pair.

Local hyperdegree D	Pair comparisons D^2	Runtime (ms)
8	64	0.05 ± 0.01
16	256	0.16 ± 0.02
32	1,024	0.59 ± 0.04
64	4,096	2.25 ± 0.09
128	16,384	8.84 ± 0.31
256	65,536	35.10 ± 1.02
512	262,144	139.82 ± 3.91

Once pairwise comparisons dominate, doubling D produces an approximately fourfold increase in Table 6, as predicted by Eq. A.1. The absolute cost remains bounded over the tested range because the descriptors are computed once and cached rather than recomputed in every training epoch.

C.3 Gate Stability under Topology Variation

We inspect the gate values produced by the trained model at the topology-evolution intensities used in Section 5.2. For each test sequence, g^R and g^P are averaged over nodes and reverse-time steps. The saturation rate is the percentage of node–time gate values below 0.05 or above 0.95. Together, these measurements show the direction of the gate response and whether the gates collapse to near-binary values.

Table 7: Gate statistics under increasing topology-evolution intensity. Values are mean $\pm$ standard deviation over runs.

ρ	Mean g^R	Mean g^P	Saturation (%)
5%	0.91 ± 0.02	0.08 ± 0.01	1.6 ± 0.3
15%	0.80 ± 0.02	0.18 ± 0.02	2.1 ± 0.4
25%	0.68 ± 0.03	0.29 ± 0.02	3.0 ± 0.5
35%	0.56 ± 0.03	0.41 ± 0.03	4.4 ± 0.7

As ρ increases, the retention gate decreases and the propagation gate increases, matching their roles in Eq. 16. The saturation rate remains below 5% throughout the tested range, so most node–time updates do not collapse to hard retention or replacement decisions.

D Additional Ablation Results

Ablation of Snapshot Embedding We assess the snapshot features by removing the propagation state, timestamp, and RWPE individually. Every removal degrades performance in Table 8. RWPE produces the largest contribution on Cora, while the propagation state contributes most on Test-Bills. The full model performs best, indicating that the three features supply different propagation, temporal, and structural information.

Table 8: Ablation results of features in snapshot embedding.

Variant	Cora		Test-Bills	
	F1	AUC	F1	AUC
w/o propagation state	0.858 ± 0.008	0.829 ± 0.010	0.861 ± 0.009	0.848 ± 0.010
w/o timestamp	0.894 ± 0.006	0.887 ± 0.007	0.896 ± 0.006	0.891 ± 0.007
w/o RWPE	0.850 ± 0.009	0.806 ± 0.011	0.883 ± 0.008	0.873 ± 0.009
Full	$\mathbf{0.917 \pm 0.004}$	$\mathbf{0.919 \pm 0.004}$	$\mathbf{0.911 \pm 0.005}$	$\mathbf{0.923 \pm 0.004}$

D.1 Ablation of the Mamba Backbone

We keep the snapshot encoder, prediction head, FDS, hidden width, and training protocol fixed. Mean pooling, GRU (Cho et al., 2014), and Transformer (Vaswani et al., 2017) replace the temporal backbone, whereas Standard Mamba (Gu & Dao, 2023) processes the snapshot sequence without hypergraph aggregation or external topology gates.

Table 9: Five-run comparison of temporal backbones. Runtime is seconds per complete training run.

Cora			
Temporal backbone	F1	AUC	Time (s)
Temporal mean pooling	0.861 ± 0.008	0.842 ± 0.010	228 ± 7
GRU	0.888 ± 0.006	0.875 ± 0.007	314 ± 8
Transformer	0.887 ± 0.006	0.843 ± 0.008	2702 ± 61
Standard Mamba	0.879 ± 0.006	0.865 ± 0.007	296 ± 7
TMDS	$\mathbf{0.917 \pm 0.004}$	$\mathbf{0.919 \pm 0.004}$	368 ± 9
Test-Bills			
Temporal backbone	F1	AUC	Time (s)
Temporal mean pooling	0.854 ± 0.009	0.838 ± 0.010	302 ± 9
GRU	0.880 ± 0.007	0.868 ± 0.008	431 ± 12
Transformer	0.878 ± 0.007	0.859 ± 0.009	3583 ± 84
Standard Mamba	0.875 ± 0.007	0.861 ± 0.008	402 ± 10
TMDS	$\mathbf{0.911 \pm 0.005}$	$\mathbf{0.923 \pm 0.004}$	506 ± 13

Table 9 compares the accuracy and efficiency of the temporal backbones. TMDS achieves the highest F1-score and AUC on both datasets, while Standard Mamba is substantially faster than the Transformer. Table 2 separately isolates the contribution of topology-conditioned gating.

D.2 Interaction between Topology Gates and FDS

Topology gates modify reverse-time state transitions, and FDS supplies forward-trajectory supervision. We evaluate their interaction with a 2×2 factorial design while keeping every other component fixed.

Table 10: Five-run interaction between topology gates and FDS.

Gates	FDS	Cora		Test-Bills	
		F1	AUC	F1	AUC
×	×	0.858 ± 0.008	0.837 ± 0.009	0.851 ± 0.009	0.828 ± 0.010
✓	×	0.903 ± 0.005	0.913 ± 0.005	0.884 ± 0.006	0.874 ± 0.007
×	✓	0.884 ± 0.006	0.878 ± 0.007	0.882 ± 0.007	0.869 ± 0.008
✓	✓	$\mathbf{0.917 \pm 0.004}$	$\mathbf{0.919 \pm 0.004}$	$\mathbf{0.911 \pm 0.005}$	$\mathbf{0.923 \pm 0.004}$

Either component improves on the variant that removes both, and their combination performs best for every dataset–metric pair. The gates provide the larger standalone gain on Cora; on Test-Bills, the two components contribute comparable gains. These results agree with their different functions: topology gates adapt the reverse dynamics, and FDS constrains the inferred sources through forward consistency.

E Additional Sensitivity Results

E.1 Impact of Snapshot Interval

TMDS ranks first at every snapshot interval on all four datasets in Figure 6, with F-scores of approximately 0.89–0.95. As the interval increases from 0.05 to 0.20, performance declines modestly and remains roughly 1–4 percentage points above the strongest baseline. House and NTU vary less than Test-Bills and Cora. The unchanged ranking suggests that the temporal module retains useful transition information when observations become less frequent, although intervals beyond the tested range remain unexamined. Appendix E also reports sensitivity to diffusion noise and source ratio.

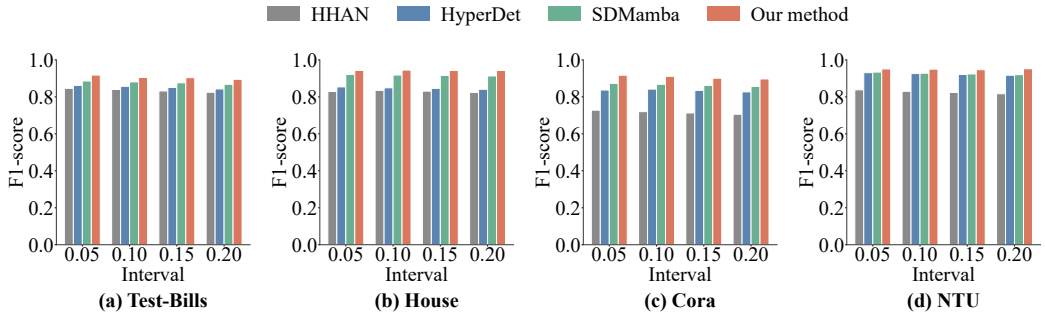


Figure 6: F-score under varying snapshot intervals on Test-Bills, House, Cora, and NTU. The diffusion-noise level and source ratio are fixed at 0.3 and 0.02, respectively.

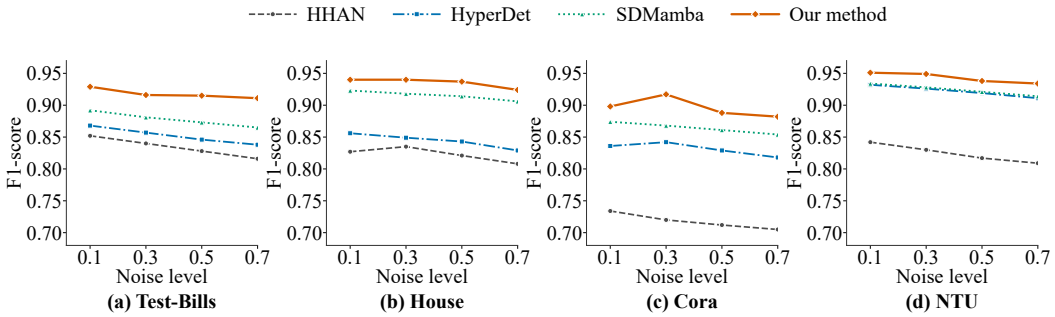


Figure 7: F-score under varying diffusion-noise levels on Test-Bills, House, Cora, and NTU. The snapshot interval and source ratio are fixed at 0.1 and 0.02, respectively.

E.2 Impact of Noise Level

TMDS achieves the highest F-score on all four datasets at every noise level in Figure 7. Performance generally decreases as noise rises from 0.1 to 0.7 because noisy infections obscure the underlying cascade structure. The decline is moderate on House, NTU, and Test-Bills, where TMDS remains approximately 2–5 percentage points above the strongest baseline at 0.7. Cora peaks at 0.3 before declining, but the method ordering does not change. The consistent lead is limited to the tested noise range, and the downward trend shows that propagation uncertainty remains a limitation.

E.3 Impact of Source Ratio

TMDS obtains the highest F-score at every source ratio on all four datasets in Figure 8. Most methods decline as the ratio increases from 0.02 to 0.08 because multiple cascades produce less distinguishable source signatures. At 0.08, TMDS exceeds the strongest baseline by approximately 2–3 percentage points on Test-Bills, Cora, and NTU and by about 1.5 points on House. Small non-monotonic variations at intermediate ratios do not change the ordering within the tested range.

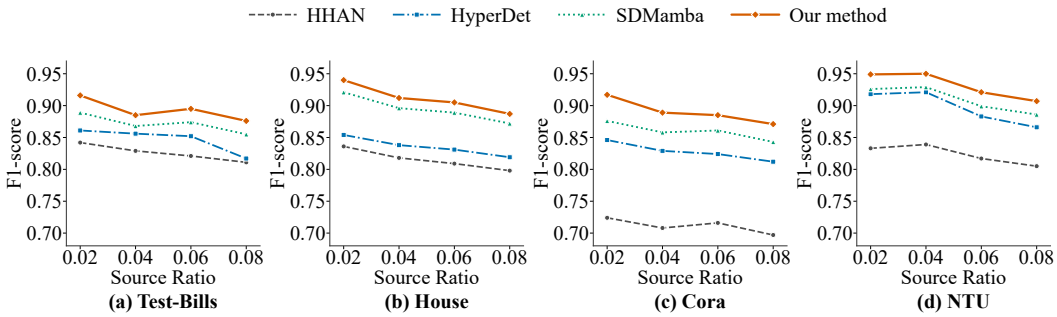


Figure 8: F-score under varying source ratios on Test-Bills, House, Cora, and NTU. The diffusion-noise level and snapshot interval are fixed at 0.3 and 0.1, respectively.