Robustness Evaluation of Graph-based News Detection Using Network Structural Information

Xianghua Zeng
zengxianghua@buaa.edu.cn
State Key Laboratory of Software Development Environment, Beihang University
Beijing, China

Hao Peng*
penghao@buaa.edu.cn
State Key Laboratory of Software Development Environment, Beihang University
Beijing, China

Angsheng Li*
angsheng@buaa.edu.cn
State Key Laboratory of Software Development Environment, Beihang University
Beijing, China

Abstract

Although Graph Neural Networks (GNNs) have shown promising potential in fake news detection, they remain highly vulnerable to adversarial manipulations within social networks. Existing methods primarily establish connections between malicious accounts and individual target news to investigate the vulnerability of graph-based detectors, while they neglect the structural relationships surrounding targets, limiting their effectiveness in robustness evaluation. In this work, we propose a novel **S**tructural **I**nformation principles-guided **A**dversarial **A**ttack **F**ramework, namely **SI2AF**, which effectively challenges graph-based detectors and further probes their detection robustness. Specifically, structural entropy is introduced to quantify the dynamic uncertainty in social engagements and identify hierarchical communities that encompass all user accounts and news posts. An influence metric is presented to measure each account's probability of engaging in random interactions, facilitating the design of multiple agents that manage distinct malicious accounts. For each target news, three attack strategies are developed through multi-agent collaboration within the associated subgraph to optimize evasion against black-box detectors. By incorporating the adversarial manipulations generated by SI2AF, we enrich the original network structure and refine graph-based detectors to improve their robustness against adversarial attacks. Extensive evaluations demonstrate that SI2AF significantly outperforms state-of-the-art baselines in attack effectiveness with an average improvement of 16.71%, and enhances GNN-based detection robustness by 41.54% on average.

Keywords

Social Networks; Fake News Detection; Structural Information

ACM Reference Format:

Xianghua Zeng, Hao Peng, and Angsheng Li. 2025. Robustness Evaluation of Graph-based News Detection Using Network Structural Information. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX).* ACM, New York, NY, USA, 13 pages. https://doi.org/XXXXXXX.XXXXXXX

*Corresponding authors.

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Recent studies [3, 32] have highlighted that the widespread growth of social media has accelerated the dissemination of misinformation and fake news. This phenomenon not only undermines public trust but also has detrimental effects on critical domains such as politics [1, 5], economics [6], and public safety [34]. Unlike traditional news articles, fake content on social platforms poses unique challenges due to its intentionally misleading nature, rapid spread, and high costs associated with expert verification [25], which necessitates the development of automated detection mechanisms. Traditional machine learning detectors [26, 30] employing natural language processing techniques aim to identify fake content to curb the spread of misinformation online. However, these approaches face efficiency limitations, especially in their capacity to account for the unique dispersion structures and complex spreading behaviors of misinformation [2]. To bridge this gap, Graph Neural Network (GNN)-based detectors [18, 21] have emerged, providing a more precise analysis of the intricate structural patterns in rumor dissemination, thereby notably enhancing detection accuracy.

Despite advancements in GNN techniques, current graph-based detectors remain susceptible to adversarial manipulations [4]. While recent work has extensively explored the resilience of NLP-based detectors [9, 10], the robustness of graph-based detectors remains largely under-researched. The Malcom framework was developed to systematically probe and exploit vulnerabilities in advanced fake news detection systems through the generation of adversarial comments [12]. Additionally, a reinforcement learning-based attack strategy was designed to identify specific weaknesses in sophisticated graph-based rumor detectors [19]. The gradient-based GAFSI framework [41] was introduced to enable general adversarial attacks against black-box detectors across various graph structures. Recognizing important social interactions and diverse fraudster types, a multi-agent reinforcement learning adversarial attack framework was proposed, employing three types of fraudsters to investigate the vulnerabilities of graph-based detectors in adversarial scenarios thoroughly [35]. Nevertheless, these methods primarily focus on associating malicious accounts with individual target news, overlooking the underlying structural relationships within the social network, which play a pivotal role in the propagation of misinformation.

This work integrates the network's structural information into SI2AF, a comprehensive attack framework designed to enhance the understanding of misinformation dynamics and assess the robustness of graph-based detectors. Initially, we extract user accounts and news posts from historical engagements to construct a bipartite user-post graph, where interactions between users and posts are

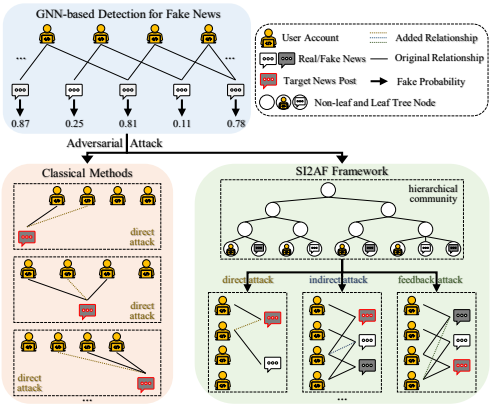

Figure 1: Comparative illustration between classical methods and our framework. SI2AF minimizes dynamic uncertainty in social engagements and identifies an associated subgraph to strategically establish connections with both target and non-target posts, resulting in significantly enhanced attack effectiveness.

modeled as random walks among graph vertices, with their dynamic uncertainties quantified through structural entropy. Subsequently, we minimize the high-dimensional structural entropy of this graph to identify a hierarchical community structure for all vertices, referred to as the optimal encoding tree. Each tree node corresponds to a vertex community, and an associated subgraph captures the frequent engagements between user and post vertices within this community. Furthermore, we present an influence metric derived from structural entropy to measure the likelihood of each account's participation in random engagements within the bipartite graph, reflecting its potential impact on the information flow. Building on this metric, we categorize user accounts into genuine accounts and distinct malicious groups with varying levels of influence, each managed by a separate decision-making agent. For each target post, we develop three attack strategies through multi-agent collaboration within the associated user-post subgraph, aiming to maximize its evasion against black-box detectors. Figure 1 illustrates a comparative analysis of attacks against a graph-based fake news detector using traditional methods and our SI2AF framework. By incorporating the generated manipulations, we enrich the structural relationships within the user-post graph and further refine graph-based models to improve detection robustness against adversarial attacks. Comprehensive experiments conducted on two real-world datasets demonstrate that SI2AF significantly outperforms state-of-the-art baselines regarding attack effectiveness and effectively enhances the robustness of graph-based detection. Our contributions can be summarized as follows:

- A novel adversarial attack framework that leverages network structural information is proposed to effectively obfuscate graph-based news detectors and evaluate their robustness.
- An influence metric is presented to quantify the likelihood of user accounts' participation in random engagements, enabling the design of multiple agents that manage distinct malicious accounts.
- Three attack strategies based on multi-agent coordination within bipartite user-post subgraphs are developed to optimize evasion against graph-based detectors.
- Comparative evaluations demonstrate that SI2AF significantly improves attack effectiveness by 16.71% and enhances graph-based detection robustness by 41.54% at average.

2 Related Work

2.1 Adversarial Attack on Graph Neural Networks

Various adversarial techniques have been developed to attack graph-based detectors via edge perturbation, aiming to probe their robustness. Nettack [42] introduced the first targeted attack utilizing incremental computation and greedy-based edge perturbations, optimizing the attack strategy step by step. SGA [16] improved attack efficiency on large-scale graphs by incorporating a subgraph construction process to misclassify targeted nodes. The minimum-budget topology attack strategy [40] was designed to determine the smallest amount of perturbation necessary to compromise each node successfully. EA-PGD [27] introduced transferable adversarial attacks to perform edge perturbations on heterogeneous graph structures. Despite their potential successes, these methods disrupt the propagation structure, resulting in insufficiency when attacking social news detectors. Recent advances [35] have led to developing a multi-agent coordination framework on three types of malicious accounts to disrupt GNN-based fake news detection. The gradient-based GAFSI method [41] has successfully executed general adversarial attacks against detectors across various graph structures. However, these approaches typically use malicious accounts to engage with individual target posts, neglecting to account for the structural relationships between posts, which limits their attack strategies and reduces the effectiveness of the attack. In contrast, our work novelly integrates structural information principles to design a range of subgraph-based attack strategies that are both more nuanced and effective than previous methods. The attack strategies outlined above are summarized in Table 1, which evaluates whether they target social news detection, model distinct groups of malicious accounts, and consider the structural relationships among target posts.

Table 1: Summary of attacks against GNN-based detectors.

Attack Method	News Detection	Distinct Group	Structural Relationship
Nettack [42]	×	×	×
SGA [16]	×	×	✓
MiBTack [40]	×	×	×
EA-PGD [27]	×	×	×
MARL [35]	✓	✓	×
GAFSI [41]	✓	×	×
SIASF (Ours)	✓	✓	✓

2.2 Structural Information Principles

In 2016, a significant advancement was made by introducing structural information, including structural entropy and partitioning trees, as proposed by [13]. This innovative concept facilitated the measurement of network complexity, laying the foundation for identifying hierarchical communities within complex systems. Building on these principles, researchers minimize high-dimensional structure entropy to classify cancer cell subtypes [14] and decode topologically associating domains within Hi-C data [15]. With further advancements, community-based structural entropy [17] emerged as a targeted measure to quantify community information aimed at solving deception-related challenges. In 2022, structural entropy's application was extended to node classification [37], resulting in the creation of SEP, which utilizes structural entropy to tackle challenges associated with local structural damage. Recent research efforts [38, 39] have concentrated on developing efficient decision-making algorithms within partitioning trees of state or action spaces.

3 Preliminaries

In this section, we begin with a definition of GNN-based fake news detection, followed by a description of adversarial attacks targeting these detectors, and conclude with an introduction to the structural information principles.

3.1 GNN-based Fake News Detection

A user-post graph, denoted as G_{up}, is a bipartite graph defined by the tuple $\{U, P, E_{up}, X_u, X_p, Y\}$, where $U = (u_0, \ldots, u_m)$ represents the set of users and $P = (p_0, \ldots, p_n)$ denotes the set of news posts. The feature matrices for users and posts are represented by X_u and X_p, respectively. The label set Y contains post labels, where 1 signifies fake news and 0 indicates real news. An edge $e_{ij} = (u_i, p_j) \in E_{up}$ implies social engagement between user u_i and post p_j.

Within a standard detection framework, a graph neural network (GNN) f_θ processes G_{up} by recursively aggregating information from neighboring vertices to obtain a representation h_p for each post p. To classify a given news post $p \in P$, the GNN representation h_p is input into a classifier f, which maps h_p to a predicted label $\hat{y} \in (0, 1)$. The cross-entropy loss for P is formulated as follows:

$$\mathcal{L}_{GNN}\left(G_{up}, f_\theta\right) = \sum_{p_i \in P}\left[-\log\left(y_i \cdot \sigma\left(f_\theta\left(X_p, E_{up}\right)_{p_i}\right)\right)\right], \quad (1)$$

where σ denotes the sigmoid function used for the binary classification of news posts.

3.2 Attacks against GNN-based News Detectors

An adversarial attack against GNN-based detectors aims to alter the classification outcomes of target news posts $P_t \subset P$ by leveraging malicious accounts $U_m \subset U$ that disseminate new posts. In this study, the GNN model f_θ is initially trained on a clean dataset, and it is assumed that the model's parameters remain unknown during the attack process. The attack objective is to maximize the misclassification rate among P_t, which is expressed as follows:

$$\begin{aligned} &\max_{E'_{up}} \sum_{p_i \in P_t} \mathbf{1}\left(f_{\theta^*}\left(X_p, E'_{up}\right)_{p_i} \neq y_i\right), \\ &\text{s.t.} \quad \theta^* = \arg\min \mathcal{L}_{GNN}\left(G_{up}, f_\theta\right), \quad |U_m| \leq \Delta_u, \end{aligned} \quad (2)$$

where E'_{up} represents the manipulated edges, and Δ_u denotes the attack budget, defined as the maximum number of controlled users.

3.3 Structural Information Principles

In an undirected graph $G = (V, E)$, a disjoint partition of all vertices is denoted as $V_0, V_1, \ldots$, with each subset V_i representing as a vertex community. These primary communities can be further subdivided into smaller sub-communities, forming a hierarchical community structure. The concept of structural entropy [13] quantifies the dynamic uncertainty encountered during a random walk between vertices within this hierarchical structure.

In the absence of a hierarchical community structure, the one-dimensional structural entropy $H^1(G)$ of the graph G is analogous to Shannon entropy [28] and is calculated based on the distribution of vertex degrees d_v as follows:

$$H^1(G) = -\sum_{v \in V} d_v \cdot \log d_v. \quad (3)$$

In this work, we define the hierarchical community structure used in SI2AF as an encoding tree with the following properties: 1) The root node λ corresponds to the entire vertex set V, such that $V_\lambda = V$. 2) Each leaf node v corresponds to an individual vertex $v \in V$, with $V_v = \{v\}$. 3) Each intermediate node α (neither root nor leaf) corresponds to a subset of vertices V_α, and its parent node is marked as α^-. 4) For each non-leaf node α, the number of its child nodes is assumed as l_α, and its i-th child is specified as $\alpha^{\langle i \rangle}$. The subsets $V_{\alpha^{\langle i \rangle}}$ are mutually exclusive and collectively exhaustive, satisfying $V_\alpha = \bigcup_{i=1}^{l_\alpha} V_{\alpha^{\langle i \rangle}}$ and $V_{\alpha^{\langle i \rangle}} \cap V_{\alpha^{\langle j \rangle}} = \emptyset$ for any $i \neq j$.

The encoding tree T significantly reduces the dynamical uncertainty within the graph G, and the high-dimensional structural entropy quantifies the residual uncertainty. The entropy assigned to a non-root node α represents the uncertainty associated with a random walk transitioning from the parent community V_{α^-} to the child community V_α, as detailed as follows:

$$H^T(G; \alpha) = -\frac{g_\alpha}{\mathcal{V}_\lambda} \log_2 \frac{\mathcal{V}_\alpha}{\mathcal{V}_{\alpha^-}}, \quad (4)$$

where $\mathcal{V}_\alpha$ is the volume of V_α, $\mathcal{V}_\alpha = \sum_{v \in V_\alpha} d_v$. The item g_α denotes the cumulative weight of all edges connecting vertices within V_α to vertices outside V_α. The K-dimensional structural entropy is defined as follows:

$$H^T(G) = \sum_{\alpha \in T, \alpha \neq \lambda} H^T(G; \alpha), \quad H^K(G) = \min_T \left\{H^T(G)\right\}, \quad (5)$$

where T ranges over all encoding trees with heights at most $K > 1$.

4 Methodology

In this work, we leverage structural information in social networks to identify the hierarchical community structure encompassing user accounts and news posts and further utilize multi-agent coordination to achieve effective attacks against GNN-based news detectors. As illustrated in Figure 2, our SI2AF framework consists of three primary modules: hierarchical structure identification, multiple agent design, and target subgraph attack. During the structure identification module, we construct a bipartite user-post graph from historical engagements and generate its optimal encoding tree, representing the hierarchical community structure of all users and posts. In the agent design module, we present an influence metric using structural entropy to evaluate user accounts, categorizing them into distinct types of malicious and genuine accounts. We coordinate multiple agents for each target post to establish new connections with both target and non-target posts within the associated subgraph, aiming to optimize evasion under GNN-based detection models.

4.1 Hierarchical Structure Identification

In contrast to previous studies [35, 41], which independently analyze individual target news, we minimize the dynamic uncertainty in social engagements to identify a hierarchical community structure of social accounts and news posts, thereby facilitating effective subgraph attacks within the SI2AF framework.

To this end, we begin by extracting historical engagements between user accounts U and news posts P to construct an undirected bipartite user-post graph $G_{up} = (U, P, E_{up})$. Following the methodology described by [7], we employ a pre-trained language model [24] to obtain user representations X_u by embedding their historical

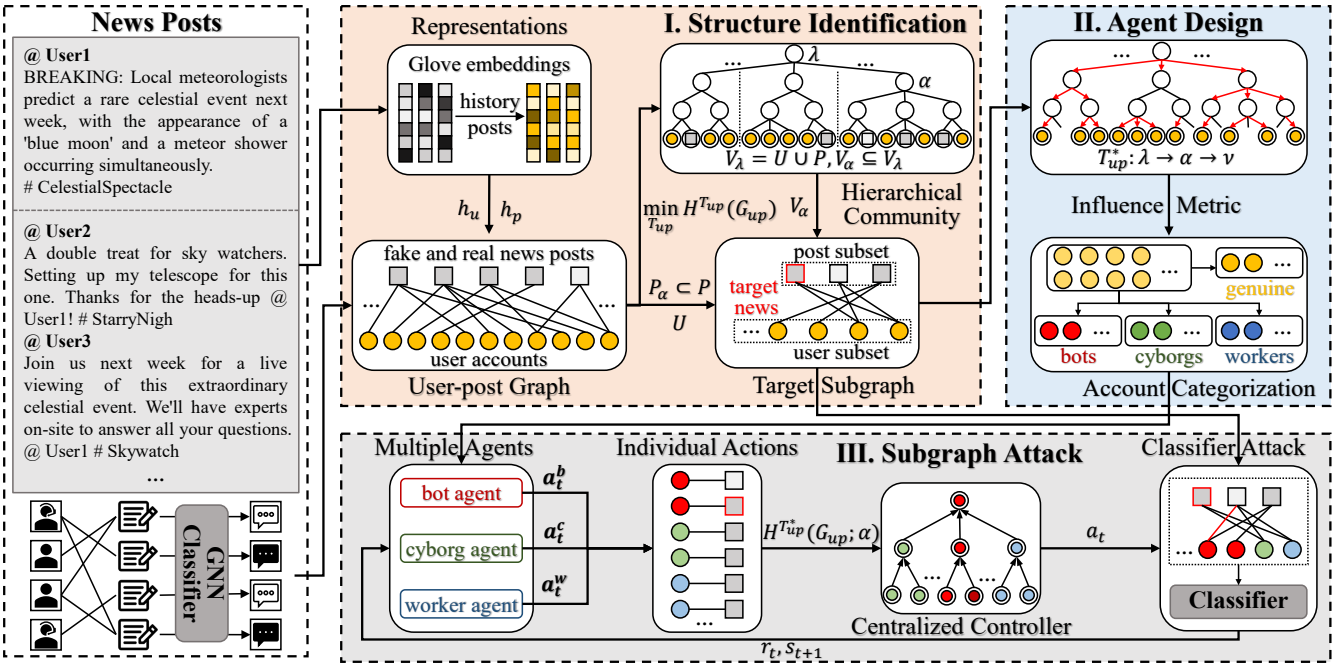

Figure 2: Detailed design of our proposed SI2AF framework.

posts. Similarly, we create post representations X_p by embedding the content of each news post.

For each edge $e_{ij} = (u_i, p_j) \in E_{up}$, we calculate the cosine similarity between the representations $h_{u_i} \in X_u$ and $h_{p_j} \in X_p$, a standard measure for capturing semantic similarity in embedding spaces. The resulting similarity score is used to compute the edge weight $w_{ij} \in [0, 1]$ as follows:

$$w_{ij} = \frac{1}{2}\left(\cos\left(h_{u_i}, h_{p_j}\right) + 1\right). \tag{6}$$

Intuitively, a higher weight w_{ij} indicates greater relevance between user u_i and post p_j, whereas a lower weight reflects dissimilarity.

In the bipartite graph G_{up}, we then model social engagements as random walks between user and post vertices, using structural entropy to quantify the dynamic uncertainty of these interactions. This entropy quantifies the minimum amount of information (in bits) required to determine accessible users or posts during a random social engagement. By minimizing the high-dimensional entropy of G_{up}, we generate its optimal encoding tree, which captures the hierarchical community structure of user accounts U and news posts P. We start by initializing a single-layer encoding tree T_{up} for G_{up}, where each leaf node v has the tree root λ as its parent, denoted as $v^- = \lambda$. Using the HCSE algorithm [23], we apply two operators, *stretch* and *compress*, to iteratively and greedily optimize the encoding tree T_{up} from a single layer to K layers, ultimately yielding the optimal K-layer encoding tree T^*_{up}. In the tree T^*_{up}, the root node λ corresponds to the union of user and post sets, $V_\lambda = U \cup P$. Each leaf node v corresponds to a singleton containing an individual user or post, while intermediate nodes correspond to communities at various hierarchical levels.

Finally, for each target post $p \in P_t$, we extract its corresponding k-layer community V_α, compressing the user subset $U_\alpha \subset U$ and post subset $P_\alpha \subset P$ at the k-th hierarchical level in T^*_{up}. We extend the user subset U_α to include the entire set U, and derive the associated bipartite subgraph $G_\alpha = (U, P_\alpha, E^{up}_\alpha)$. The extended vertex subset consists of two components: the entire account set U and the post subset $P_\alpha \subset P$. The edge subset E^{up}_α captures the local structural relationships between the accounts in U and posts in P_α, highlighting their interactions within the subgraph.

In this work, we set the height parameter k as $K - 1$ by default, enabling us to derive all targeted subgraphs from the vertex communities corresponding to the immediate children of the root node.

4.2 Multiple Agent Design

Building on this hierarchical community structure, we present a metric to measure each user account's network influence and design multiple cooperative agents to manage malicious accounts, taking into account their distinct influences and budgets.

In the encoding tree T^*_{up}, the structural entropy assigned to each non-root node α in Equation 4 measures the uncertainty of a random walk transitioning from the parent community V_{α^-} to its child community V_α. For any user $u \in U$, the probability of a random engagement reaching this user is determined by the cumulative entropy of all nodes α encountered along the path from the root node λ to the leaf node v, where $V_v = \{u\}$. Consequently, we define the influence metric $\mathcal{I}$ as a measure of each user account's likelihood of engaging in random interactions within G_{up} as detailed below:

$$\mathcal{I}(G_{up}; u) = \sum_{V_v \subseteq V_\alpha \subset V} \left[-\frac{g_\alpha}{\mathcal{V}_\lambda} \log_2 \frac{c \cdot \mathcal{V}_\alpha}{\mathcal{V}_{\alpha^-}} \right], \tag{7}$$

where c serves as an adjusting parameter that modulates the distribution of influence across all user accounts.

Prior research [35] has categorized distinct malicious groups according to the number of news shares per account, which indicates

Algorithm 1: Malicious Accounts Categorization

Input: user account set U, bot budget Δ_b, cyborg budget Δ_c, worker budget Δ_w

Output: bot set U_b, cyborg set U_c, worker set U_w

1 $U' \leftarrow$ sort U by network influence using Equation 7
2 $\Delta \leftarrow \Delta_b + \Delta_c + \Delta_w$
3 $U_l \leftarrow U'[: \lfloor \frac{\Delta_b}{\Delta} \rfloor]$
4 $U_m \leftarrow U'[\lfloor \frac{\Delta_b}{\Delta} \rfloor : \lfloor \frac{\Delta_b}{\Delta} \rfloor + \lfloor \frac{\Delta_c}{\Delta} \rfloor]$
5 $U_h \leftarrow U'[\lfloor \frac{\Delta_b}{\Delta} \rfloor + \lfloor \frac{\Delta_c}{\Delta} \rfloor :]$
6 $U_b \leftarrow$ randomly sample Δ_b accounts from U_l
7 $U_c \leftarrow$ randomly sample Δ_c accounts from U_m
8 $U_w \leftarrow$ randomly sample Δ_w accounts from U_h

their network influence. However, due to the sparsity of social networks, where most users are linked to only a single news post, this leads to a heavy-tailed sharing distribution, which can cause imbalances in influence-based categorizations. By integrating content relevance and hierarchical community structure into our metric, we achieve a more nuanced differentiation between users who share the same number of posts, enhancing the precision of influence measurement compared to previous methods. The following theorem demonstrates that, even in the context of an unweighted graph and a single-layer network structure, adjusting the parameter c within the influence metric can reduce the occurrence of accounts with identical influence values, thus fostering a more balanced distribution of user influence. In this particular scenario, the influence metric $\mathcal{I}(G_{up}; u)$ for each user $u \in U$ depends exclusively on the user's vertex degree, which represents the number of connections the user has to different pieces of content. This influence metric is formally defined as follows:

$$\mathcal{I}(G_{up}; u) = -\frac{g_v}{\mathcal{V}_\lambda} \log_2 \frac{c \cdot \mathcal{V}_v}{\mathcal{V}_\lambda} = -\frac{d_u}{\mathcal{V}_\lambda} \log_2 \frac{c \cdot d_u}{\mathcal{V}_\lambda}. \tag{8}$$

THEOREM 4.1. *Let $x \in [1, \frac{b}{2}]$ be a positive random variable with a probability density function $q_0(x)$. Given the transformation $x' = -\frac{x}{b} \cdot \left(\log_2 \frac{c}{b} x\right)$, under the condition $0 < c \leq \frac{2}{e}$, the variable x' increases monotonically with the variable x, and its probability density function $q_1(x')$ satisfies:*

$$0 \leq q_1(x') \leq \frac{b}{1 - \log_2 ec}. \tag{9}$$

A detailed proof of this theorem is provided in Appendix B. The parameter b denotes the sum of the total number of posts shared by all users and the total number of times these posts have been shared. As a result, each user's individual sharing count x is bounded by the range of $1 \leq x \leq \frac{b}{2}$.

Given the involvement of various malicious groups in misinformation campaigns [22, 29], we model three distinct types of malicious accounts—bots, cyborgs, and crowd workers—characterized by varying influence levels and different budgets. Using the budgets for the three malicious groups, denoted as Δ_b, Δ_c, and Δ_w, we develop an adaptive categorization algorithm that generates the bot group U_b with low influence, the cyborg group U_c with medium influence, and the worker group U_w with high influence. Specifically, we first sort all users in U by the influence metric $\mathcal{I}$ in ascending order (line 1 in Algorithm 1). Next, we determine the sizes of the low, medium, and high influence groups according to the specified budgets and categorize all accounts into these groups (lines 2-5 in Algorithm 1). Finally, we randomly sample accounts from each group to return U_b, U_c, and U_w (lines 6-8 in Algorithm 1). The controlled malicious accounts U_m are thus defined as follows:

$$U_m = U_b \cup U_c \cup U_w. \tag{10}$$

Finally, to simulate the coordinated behavior among the different groups, we design three agents, each embodying a distinct level of influence: agent $\mathcal{N}_b$ for low-influence social bots, agent $\mathcal{N}_c$ for medium-influence cyborg accounts, and agent $\mathcal{N}_w$ for high-influence crowd workers.

4.3 Target Subgraph Attack

Each attack on a target post, primarily focusing on fake news (though applicable to real news as well), is modeled as a collective effort within the associated user-post subgraph, where all agents work collaboratively to manipulate the classification outcome of a black-box GNN-based detector.

For a target fake news post $p \in P_t$, its associated user-post subgraph G_α includes the closely related posts P_α, including both fake news posts $P_\alpha^f = \{p_1^f, p_2^f, \ldots, p_{l_f}^f\}$ with $p = p_1^f$ and real news posts $P_\alpha^r = \{p_1^r, p_2^r, \ldots, p_{l_r}\}$. Here, l_f and l_r denote the number of fake and real news posts, respectively, within the post subset $P_\alpha \subset P$.

The SI2AF framework models the attack on the target news post p as a cooperative multi-agent Markov decision process, characterized by the tuple $(\mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{N} = \{\mathcal{N}_b, \mathcal{N}_c, \mathcal{N}_w\}$ is the set of agents, $\mathcal{S}$ denotes the state space observed by all agents, $\mathcal{A}$ is the joint action space, $\mathcal{P}$ represents the transition function, $\mathcal{R}$ refer to the reward function, and γ is the discount factor. At each timestep t, the agent $\mathcal{N}_b$, responsible for managing malicious accounts $U_b = \{u_1^b, u_2^b, \ldots, u_{\Delta_b}^b\}$, observes the current environmental state $s_t \in \mathcal{S}$ and selects actions $\boldsymbol{a_t^b} = (a_1^b, a_2^b, \ldots, a_{\Delta_b}^b)$ according to its policy network π_b. The policy network π_b determines which post vertex each controlled account will interact with in the associated user-post subgraph G_α, expressed as $\boldsymbol{a_t^b} = \pi_b(s_t, U_b, G_\alpha)$. Similarly, agents $\mathcal{N}_c$ and $\mathcal{N}_w$ select their respective actions $\boldsymbol{a_t^c}$ and $\boldsymbol{a_t^w}$ using their own policy networks, following a decision-making process analogous to that of $\mathcal{N}_b$.

For each malicious account $u_i^b \in U_b$, we define the sampled probability p_i^b of its selected action a_i^b based on the cumulative entropy of all common parent nodes shared by u_i^b and the target post $p \in P$ as follows:

$$p_i^b = \sum_{\{u_i^b, p\} \subset V_\alpha} H^{T_{up}^*}(G_{up}; \alpha). \tag{11}$$

If the only common parent node between u_i^b and p is the root node, we set the sampled probability p_i^b to a predefined random small value, 0.01, to reflect a low likelihood of action. Similarly, we define the sampled probabilities for the accounts controlled by agents $\mathcal{N}_c$ and $\mathcal{N}_w$ following the same approach. Based on these probabilities, we perform a weighted sampling process on $\boldsymbol{a_t^b}$, $\boldsymbol{a_t^c}$, and $\boldsymbol{a_t^w}$, which leads to the single-agent actions a_t^b, a_t^c, and a_t^w at timestep t. Moreover, we centrally aggregate these actions, weighting them according to

the sum of the network influences exerted by the malicious accounts controlled by each agent. This aggregation yields the final action a_t at timestep t, which specifies the attacked post $p_t \in P_\alpha$ and the selected malicious account $u_t \in U_m$. The collective action (p_t, u_t) modifies the structure of user-post graph G_{up} by establishing a new sharing between u_t and p_t, potentially affecting the classification outcome of target news p by the GNN-based detector f_{θ^*}. Depending on the types of attacked post p_t, our subgraph attack encompasses three distinct attack strategies:

• **Direct Attack**: Directly interact with the target news, $p_t = p$, to affect its classification outcome by the GNN-based detector.

• **Indirect Attack**: Engage with real news within the associated subgraph, $p_t \in P^r_\alpha$, to indirectly affect the prediction of target p.

• **Feedback Attack**: Interact with other fake news in the associated subgraph, $p_t \in P^f_\alpha$ and $p_t \neq p$, with the aim to enrich the environmental feedback and address the challenge of reward sparsity in the decision process.

In its efforts to mount adversarial attacks against the GNN-based detectors f_{θ^*}, the SI2AF framework considers the classification outcomes of the target post and other related fake news posts. The predictions $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{l_f}$ for these posts act as reward signals, guiding the training and refinement of the policy networks of all agents in $\mathcal{N}$. Specifically, $\hat{y}_1 = f_{\theta^*}\left(X_p, E'_{up}\right)_{p^f_1}$ represents the classification outcome for the target post $p = p^f_1$, while $\hat{y}_i = f_{\theta^*}\left(X_p, E'_{up}\right)_{p^f_i}$ denotes the results for other fake posts p^f_i where $i > 1$. The reward $\mathcal{R}(s_t, a_t)$ is defined as follows:

$$\mathcal{R}(s_t, a_t) = \begin{cases} 1 & \text{if } f_{\theta^*}\left(X_p, E'_{up}\right)_{p^f_1} \neq y_1, \\ \sum_{i=2}^{l_f} \frac{\mathbf{1}\left(f_{\theta^*}\left(X_p, E'_{up}\right)_{p^f_i} \neq y_i\right)}{l_f - 1} & \text{otherwise,} \end{cases} \tag{12}$$

where E'_{up} denotes the updated structural relationships perturbed by the action a_t.

For the agent $\mathcal{N}_b$ operating under policy π_b, we employ Q-learning to estimate its value function Q_b and minimize the optimization loss as follows:

$$\mathcal{L}_{Q_b} = \mathbb{E}_{(s_t, a^b_t)}\left[\mathcal{R}(s_t, a^b_t) + \gamma \max_{a^b_{t+1}} Q^-_b(s_{t+1}, a^b_{t+1}) - Q_b(s_t, a^b_t)\right], \tag{13}$$

where Q^-_b denotes the target value network of agent $\mathcal{N}_b$, introduced to stabilize the training process by reducing oscillations in the learned Q-values. The optimal value function $Q^*_b(s_t, a^b_t)$ is expressed using the following Bellman Equation:

$$Q^*_b(s_t, a^b_t) = \mathcal{R}(s_t, a^b_t) + \gamma \max_{a^b_{t+1}} Q^*_b(s_{t+1}, a^b_{t+1}). \tag{14}$$

This equation describes a greedy policy, where the agent $\mathcal{N}_b$ selects the action that maximizes the Q-value for the given state:

$$\pi_b(\boldsymbol{a^b_t} | s_t; Q^*_b) = \arg\max_{\boldsymbol{a^b_t}} Q^*_b(s_t, \boldsymbol{a^b_{t+1}}). \tag{15}$$

The policy training for agents $\mathcal{N}_c$ and $\mathcal{N}_w$ follows the same Q-learning methodology as that of $\mathcal{N}_b$, with adaptations to their unique action spaces.

Algorithm 2: Graph-based Detector Optimization

Input: target post set P_t, trained policies $\pi^*_b, \pi^*_c, \pi^*_w$
Output: optimized detector f_{θ^*}

1 $E'_{up} = E_{up}$
2 **for** $p \in P_t$ **do**
3 $\quad G_\alpha = (U, P_\alpha, E^{up}_\alpha) \leftarrow$ derive the associated subgraph
4 $\quad P^f_\alpha$ and $P^r_\alpha \leftarrow$ extract the fake and real news in G_α
5 $\quad$ **while** $t < t_{max}$ **do**
6 $\quad\quad \boldsymbol{a^b_t} \leftarrow \pi^*_b(s_t, U_b, G_\alpha)$
7 $\quad\quad \boldsymbol{a^c_t} \leftarrow \pi^*_c(s_t, U_c, G_\alpha)$
8 $\quad\quad \boldsymbol{a^w_t} \leftarrow \pi^*_w(s_t, U_w, G_\alpha)$
9 $\quad\quad a^b_t$, a^c_t, and $a^w_t \leftarrow$ individually sample $\boldsymbol{a^b_t}$, $\boldsymbol{a^c_t}$, and $\boldsymbol{a^w_t}$ via Equation 11
10 $\quad\quad a_t = (u_t, p_t) \leftarrow$ sample single-agent actions a^b_t, a^c_t, and a^w_t
11 $\quad\quad E'_{up} = E'_{up} \bigcup \{(u_t, p_t)\}$
12 $f_{\theta^*} \leftarrow$ refine the graph-based detector on E'_{up} by minimizing the cross-entropy loss in Equation 1

4.4 Detection Optimization

By leveraging the trained SI2AF framework, we incorporate the generated manipulations within each subgraph to update the structural relationships between users and posts, thereby optimizing graph-based detectors to improve their robustness.

For each target post $p \in P_t$, we extract the fake news p^f_α and real news p^r_α from the associated subgraph G_α (lines 3 and 4 in Algorithm 2). According to the trained policies π^*_b, π^*_c, and π^*_w, we select the multi-agent actions $\boldsymbol{a^b_t}$, $\boldsymbol{a^c_t}$, and $\boldsymbol{a^w_t}$, respectively, and employ weighted sampling to determine the collective action $a_t = (p_t, u_t)$ at timestep t (lines 6-10 in Algorithm 2). This action is then used to update the structural relationships E_{up} within the user-post graph G_{up} (line 11 in Algorithm 2). After all attacks targeting posts in P_t, we minimize the cross-entropy loss in Equation 1 to refine the graph-based model f_{θ^*}, thereby enhancing its detection robustness.

5 Experiments

In this section, we conduct comprehensive comparative experiments on various real-world datasets to evaluate the performance of our proposed framework, SI2AF. To ensure a fair and robust assessment, the experimental results are reported as average values with standard deviations, calculated over five different random seeds.

5.1 Experimental Settings

Datasets. For our analysis, we adopt two well-established real-world datasets, Politifact and Gossipcop, which originate from two fact-checking platforms and include social interactions from Twitter [31]. These datasets contain metadata such as user interactions, post characteristics for fake and real news posts, and account information involved in these engagements. In line with existing studies [7], we utilize Glove embeddings [24] to encode both the semantic content of posts and the historical posts of users. Following the experimental settings [35], we adopt the same budget setting, randomly sampling

Table 2: The success rates of the SI2AF and other baselines on both fake and real news within the Politifact and Gossipcop datasets: "average value ± standard deviation". Bold: the best performance in each graph, underline: the second performance.

Method	Politifact Fake News					Politifact Real News				
	GAT	GCN	SAGE	Bi-GCN	GCAN	GAT	GCN	SAGE	Bi-GCN	GCAN
Random	0.14 ± 0.01	0.09 ± 0.03	0.13 ± 0.01	0.10 ± 0.02	0.09 ± 0.01	0.11 ± 0.01	0.36 ± 0.04	0.15 ± 0.01	0.16 ± 0.03	0.05 ± 0.01
DICE	0.26 ± 0.02	0.16 ± 0.01	0.24 ± 0.03	0.15 ± 0.01	0.17 ± 0.03	0.22 ± 0.03	0.38 ± 0.02	0.21 ± 0.01	0.21 ± 0.02	0.10 ± 0.02
SGA	0.32 ± 0.04	0.24 ± 0.01	0.35 ± 0.03	0.21 ± 0.01	0.27 ± 0.01	0.18 ± 0.03	0.45 ± 0.04	0.29 ± 0.01	0.32 ± 0.03	0.15 ± 0.03
GAFSI	0.35 ± 0.03	0.28 ± 0.02	0.36 ± 0.02	0.19 ± 0.02	0.26 ± 0.04	0.37 ± 0.02	0.42 ± 0.02	0.37 ± 0.03	0.41 ± 0.02	0.13 ± 0.01
MARL	0.30 ± 0.01	0.21 ± 0.01	0.35 ± 0.03	0.12 ± 0.02	0.18 ± 0.02	0.47 ± 0.01	0.35 ± 0.02	0.18 ± 0.09	0.27 ± 0.03	0.16 ± 0.02
SI2AF(Ours)	**0.41** ± 0.03	**0.31** ± 0.04	**0.41** ± 0.01	**0.28** ± 0.05	**0.34** ± 0.01	**0.69** ± 0.04	**0.56** ± 0.03	**0.49** ± 0.02	**0.45** ± 0.03	**0.19** ± 0.01
Abs.(%) Avg.↑	0.06(17.14%)	0.03(10.71%)	0.05(13.89%)	0.07(33.33%)	0.07(25.93%)	0.22(46.81%)	0.11(24.44%)	0.12(32.43%)	0.04(9.76%)	0.03(18.75%)
Method	**Gossipcop Fake News**					**Gossipcop Real News**				
	GAT	GCN	SAGE	Bi-GCN	GCAN	GAT	GCN	SAGE	Bi-GCN	GCAN
Random	0.18 ± 0.01	0.25 ± 0.03	0.15 ± 0.01	0.17 ± 0.02	0.33 ± 0.02	0.08 ± 0.01	0.17 ± 0.03	0.14 ± 0.01	0.15 ± 0.02	0.26 ± 0.02
DICE	0.13 ± 0.02	0.10 ± 0.01	0.16 ± 0.02	0.24 ± 0.03	0.29 ± 0.02	0.11 ± 0.03	0.21 ± 0.02	0.18 ± 0.02	0.27 ± 0.01	0.23 ± 0.02
SGA	0.42 ± 0.02	0.72 ± 0.03	0.19 ± 0.03	0.33 ± 0.01	0.53 ± 0.02	0.28 ± 0.01	0.45 ± 0.06	0.37 ± 0.04	0.31 ± 0.03	0.29 ± 0.02
GAFSI	0.21 ± 0.01	0.67 ± 0.04	0.20 ± 0.04	0.49 ± 0.03	0.61 ± 0.03	0.29 ± 0.03	0.43 ± 0.04	0.39 ± 0.02	0.40 ± 0.03	0.37 ± 0.04
MARL	0.80 ± 0.04	0.78 ± 0.02	0.13 ± 0.01	0.28 ± 0.05	0.78 ± 0.09	0.25 ± 0.03	0.41 ± 0.01	0.29 ± 0.03	0.44 ± 0.02	0.43 ± 0.01
SI2AF(Ours)	**0.90** ± 0.02	**0.87** ± 0.01	**0.21** ± 0.01	**0.60** ± 0.01	**0.88** ± 0.02	**0.32** ± 0.01	**0.52** ± 0.01	**0.42** ± 0.01	**0.47** ± 0.01	**0.46** ± 0.01
Abs.(%) Avg.↑	0.10(12.5%)	0.09(11.54%)	0.01(5.00%)	0.11(22.45%)	0.10(12.82%)	0.03(10.34%)	0.07(15.56%)	0.03(7.69%)	0.03(6.82%)	0.03(6.98%)

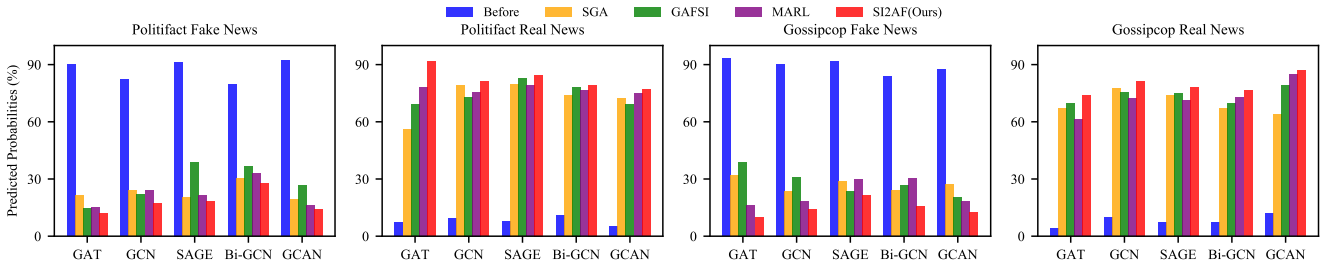

Figure 3: Average predictive probabilities of fake and real news before and after adversarial attacks.

100 bots, 50 cyborgs, and 20 crowd workers from the Politifact dataset, and 1000 bots, 500 cyborgs, and 100 crowd workers from the Gossipcop dataset.

Detection Models. In this study, we evaluate the effectiveness of SI2AF attack strategies against five different GNN-based detectors on the Politifact and Gossipcop datasets. The GNN models employed as detectors include Graph Convolution Network (GCN) [11], Graph Attention Network (GAT) [33], Graph Sample and Aggregation Network (GraphSAGE) [8], Graph-aware Co-Attention Network (GCAN) [18], and Bi-Directional Graph Convolutional Network (Bi-GCN) [2]. Their detection performances after training are provided in Appendix C.2.

Baselines. We compare SI2AF with several state-of-the-art baselines, including random-based methods (Random and DICE [36]), gradient-based methods (SGA [16] and GASFI [41]), and the multi-agent cooperative method (MARL [35]), using their publicly available open-source implementations.

5.2 Evaluation

To evaluate attack performance, we use the success rate as our primary metric, defined as the proportion of target posts successfully misclassified by the GNN-based detectors. We assess SI2AF and other baselines on their abilities to misclassify fake and real news in the Politifact and Gossipcop datasets, reporting the average success rate and standard deviation in Table 2. Our experimental results show that SI2AF consistently outperforms all baselines, achieving maximum success rate improvements of up to 33.33% for fake news and 46.81% for real news across various attack scenarios. For a deeper

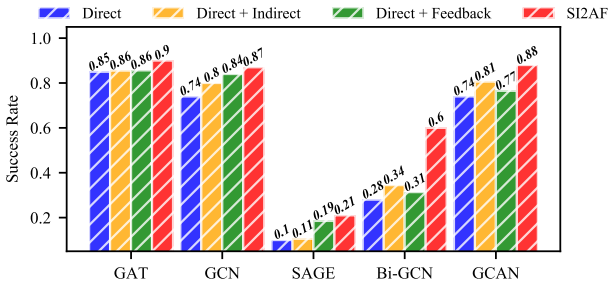

Figure 4: Success rates of different attack strategies on fake news in the Gossipcop dataset.

understanding of attack performance, we present the average predictive probabilities of target news posts in Figure 3, illustrating the likelihood of these posts being classified as fake before and after attacks. The results indicate that SI2AF induces more significant changes in predictive probabilities for fake and real posts, outperforming the best-performing baselines (SGA, GAFSI, and MARL) across different detection models. Specifically, SI2AF achieves an average reduction of 71.90% in the predictive probabilities of fake posts and an increase of 72.90% for real posts.

As outlined in Section 4.3, the subgraph attack in SI2AF incorporates three distinct strategies, each targeting a specific category of news posts. In Figure 4, we analyze the impact of selectively applying different attack strategies within the Gossipcop dataset and report the corresponding success rates. The combination of different attack strategies consistently yields higher success rates than using any single attack type alone, highlighting the strategic advantage

Table 3: The detection performance of graph-based detectors against Gossipcop fake news before and after attacks.

Method	GAT	GCN	SAGE	Bi-GCN	GCAN
Before	93.1 ± 0.5	90.4 ± 0.3	91.8 ± 0.3	83.9 ± 0.7	87.3 ± 0.4
SGA	76.3 ± 0.8	64.1 ± 0.5	75.4 ± 0.3	62.8 ± 0.3	71.6 ± 0.4
GAFSI	72.9 ± 1.0	68.3 ± 0.5	65.3 ± 0.6	65.1 ± 0.5	60.7 ± 0.4
MARL	63.0 ± 0.4	68.4 ± 0.7	70.9 ± 0.2	61.4 ± 0.6	58.2 ± 0.8
SI2AF	56.4 ± 0.6	60.5 ± 0.3	61.8 ± 0.2	53.7 ± 0.3	56.2 ± 0.5

Table 4: Efficiency comparison between SI2AF and MARL with different attack budgets.

Methods	Attack Budgets ($\Delta_b : \Delta_c : \Delta_w$)			
	100: 50: 20		150: 75: 30	
	Training Time	Inference Time	Training Time	Inference Time
MARL	493.99	29.23	548.36	52.64
SI2AF	514.32	34.57	560.41	59.16
Methods	Attack Budgets ($\Delta_b : \Delta_c : \Delta_w$)			
	200: 100: 40		250: 125: 50	
	Training Time	Inference Time	Training Time	Inference Time
MARL	579.59	63.67	586.70	71.04
SI2AF	582.35	71.47	594.28	75.33

and adaptability of the subgraph attack mechanism within SI2AF. By adaptively deriving the hierarchical community structure for all accounts and posts, the SI2AF framework enables a more comprehensive and targeted subgraph attack on closely related news posts, significantly enhancing the attack's precision and effectiveness.

In summary, our SI2AF framework leverages the structural information inherent in social networks to provide a more comprehensive and effective evaluation of robustness across various graph-based detectors. Meanwhile, as illustrated in Algorithm 2, we incorporate the generated manipulations from SI2AF to enrich the network structure and refine all five detectors on the updated network. We summarize their detection performance on the Gossipcop fake news dataset, both before and after SI2AF attacks, alongside three best-performing baselines, in Table 3. It is noted that, regardless of the graph-based detector used, the drop in predictive probability for each attack algorithm is significantly mitigated after optimization, with a reduction of at average 41.54%. The average prediction probability for fake news remains above 53.7%. This is due to the comprehensive and strategic manipulations in SI2AF, which enable the detector to anticipate structural changes in the network caused by the attack algorithms, thereby significantly mitigating their impact on detection.

To intuitively reflect the efficiency and scalability of our framework, we progressively increase the attack budget, that is, the number of malicious accounts controlled by the three agents, and record the time cost (ms) on single training or inference for both SI2AF and MARL in Table 4. Although incorporating the local structure around target news introduces additional computational overhead, the actual training and inference time of SI2AF remains comparable to that of MARL and remains stable as the budget increases, further demonstrating the practicality of our framework.

To further explore the attack performance on posts with different engagements, we conduct additional analysis to evaluate the performance of our attack framework on target fake posts with varying levels of engagement. Specifically, we examine how the average success rate of fake posts varied based on their engagement levels, from newly posted, low-engagement content to high-engagement posts that had already spread significantly, using the SAGE detector

Table 5: Attack performance of SI2AF and three best-performing baselines on posts with different engagements.

Post Degree	$[0, 10)$	$[10, 100)$	$[100,)$
SGA	0.39	0.33	0.21
GAFSI	0.38	0.35	0.17
MARL	0.38	0.32	0.24
SI2AF	0.44	0.40	0.35

on the Politifact dataset. As shown in Table 5, compared to the baselines (SGA, GAFSI, and MARL), our method consistently achieves better attack performance across posts with different engagement levels. Notably, the advantage of our approach is more pronounced in high-degree posts, where attacking is more challenging due to their widespread engagement. This improvement is attributed to the richer and more diverse set of attack strategies we introduce, which are better suited to handle posts with varying degrees of influence.

5.3 Case Study

In this subsection, we focus on a specific fake post from the Gossipcop dataset and visualize the temporal changes in the associated user-post subgraph. We examine these changes induced by two multi-agent collaboration-based attack models: MARL and SI2AF. Finally, we compare the performance of these models based on their impact on the GNN-predicted probabilities.

To better visualize this process, we ensure that both attack models control the same set of malicious accounts, which are confined within their respective user-post communities. Figure 5 illustrates that SI2AF initially selects malicious accounts in a manner similar to MARL, establishing direct connections with the target post. At this stage, SI2AF does not demonstrate a clear performance advantage over MARL in terms of modifying the GNN-predicted probability. Subsequently, SI2AF expands its attack by connecting to other false and real posts, employing diverse strategies to influence the network. This leads to significant modifications in the target post's GNN-based predicted probability, enhancing the overall effectiveness of the attack.

Further, we conduct a qualitative analysis of the SI2AF attack process, specifically focusing on how it influences the predicted probabilities of fake news in the context of the GNN-based model. We identify three primary mechanisms contributing to the increased predicted probabilities of fake news:

- Establishing new connections with influential malicious accounts, in accordance with our direct attack strategy.
- Amplifying the influence of malicious accounts already linked to the target post by connecting them to additional real news, consistent with our indirect attack strategy.
- Combining direct and indirect attacks to further increase the predicted probabilities of other fake news related to the target post, in line with our feedback strategy.

5.4 Ablation Studies

In this subsection, we conduct an ablation study on the Politifact dataset for fake news detection to evaluate the impacts of various agents and their account quantities (ranging from 50 to 300) on the attack effectiveness within the SI2AF framework. We focus on

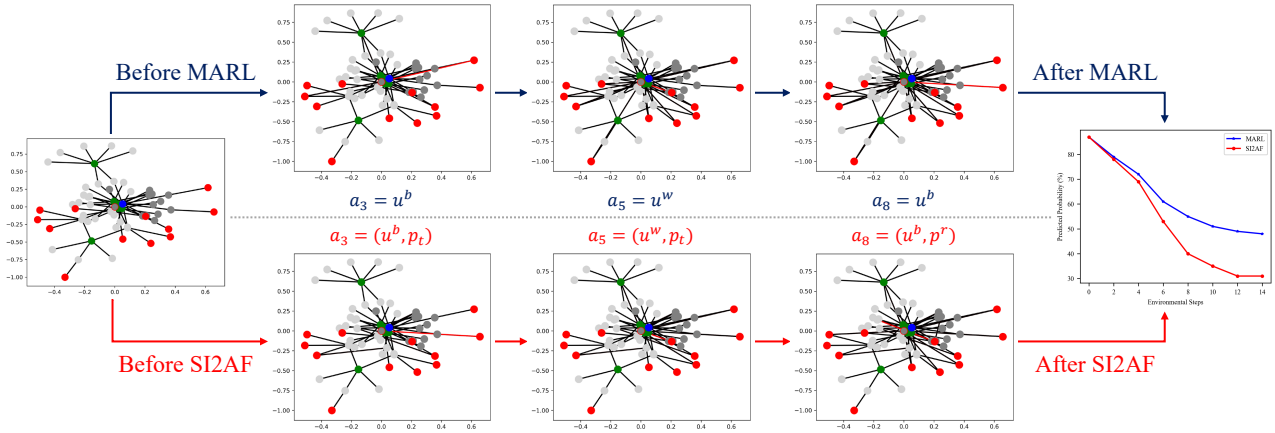

Figure 5: Attack visualization of SI2AF and MARL against GNN-based fake news detectors.

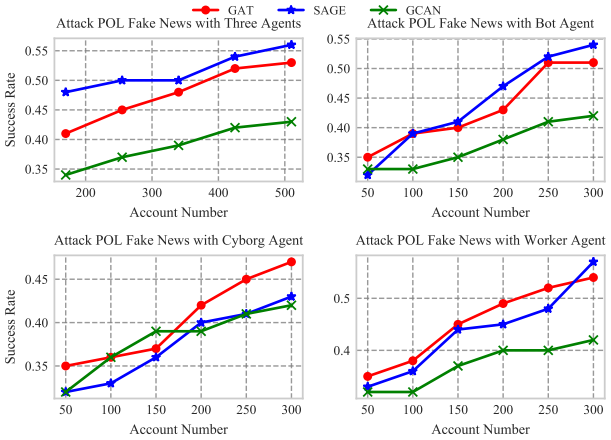

Figure 6: Attack performance of different agents on fake news detectors (GAT, SAGE, and GCAN) in the Politifact dataset.

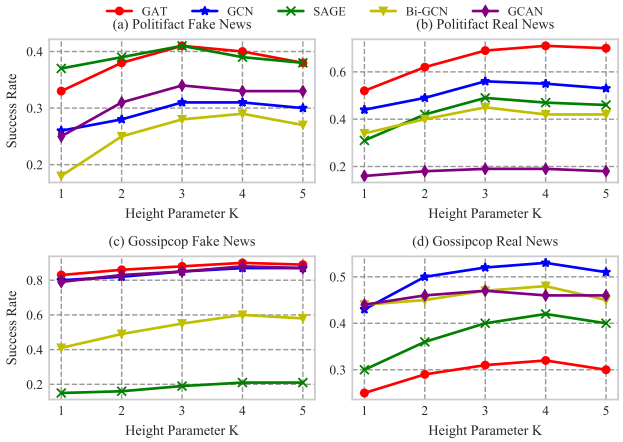

Figure 7: The success rate of SI2AF when adopting different height parameters K.

three of the most effective fake news detectors—GAT, SAGE, and GCAN—and gradually increase the number of accounts under each agent's control to carry out the attack. Our results in Figure 6 suggest that increasing the number of accounts enhances attack performance for all agents up to a certain threshold, beyond which the improvement rate levels off. Rather than improving performance, adding more accounts introduces more potential actions that may not contribute meaningfully to the attack strategy. Notably, the worker agent consistently outperforms both the bot and cyborg agents across all three GNN detectors when controlling an equal number of accounts.

5.5 Parameter Sensitivity

In this subsection, we examine the attack performance of SI2AF with different height parameters, K, which controls the size of the subgraph considered in the targeted attack. As shown in Figure 7, in both datasets, the attack success rate of SI2AF increases significantly as the value of parameter K increases. However, after reaching an optimal value, further increases in K result in a slight performance decrease. This decline occurs because a larger subgraph is more likely to include news posts unrelated to the target post, which dilutes the focus of the attack and reduces the effectiveness of SI2AF. Additionally, the optimal value of K is closely related to the social network scale. For instance, in the smaller network Politifact, SI2AF achieves its best performance when $K = 3$, while in the larger-scale network Gossipcop, peak performance occurs when $K = 4$.

6 Conclusion

This paper proposes SI2AF, an adversarial attack framework that leverages network structural information to identify the hierarchical community structure among accounts and posts, thereby facilitating effective attacks against various GNN-based detectors and evaluating their robustness. We present an influence metric for categorizing malicious accounts, combined with three subgraph strategies utilizing multi-agent collaboration to maximize target news posts' evasion. Extensive experiments on two real-world datasets, Politifact and Gossipcop, demonstrate that SI2AF consistently enhances the attack effectiveness, outperforming state-of-the-art baselines, and significantly improves the robustness of graph-based detection. Future research will focus on expanding the scope of graph-based detectors and enhancing their robustness through a more comprehensive exploration of subgraph attacks. Additionally, we plan to extend the categorization of malicious accounts and incorporate the dynamics of genuine accounts as key areas of investigation.

7 Acknowledgments

This work has been supported by NSFC through grants 62322202, 62441612 and 62432006, Local Science and Technology Development Fund of Hebei Province Guided by the Central Government of China through grant 246Z0102G, the "Pioneer" and "Leading Goose" R&D Program of Zhejiang" through grant 2025C02044, National Key Laboratory under grant 241-HF-D07-01, and Hebei Natural Science Foundation through grant F2024210008.

References

[1] Sinan Aral and Dean Eckles. 2019. Protecting elections from social media manipulation. *Science* 365, 6456 (2019), 858–861.

[2] Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, and Junzhou Huang. 2020. Rumor detection on social media with bi-directional graph convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 549–556.

[3] Canyu Chen, Haoran Wang, Matthew Shapiro, Yunyu Xiao, Fei Wang, and Kai Shu. 2022. Combating health misinformation in social media: Characterization, detection, intervention, and open issues. *arXiv preprint arXiv:2211.05289* (2022).

[4] Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. 2018. Adversarial attack on graph structured data. In *International conference on machine learning*. PMLR, 1115–1124.

[5] Ashok Deb, Luca Luceri, Adam Badaway, and Emilio Ferrara. 2019. Perils and challenges of social media and election manipulation analysis: The 2018 us midterms. In *Companion Proceedings of the 2019 World Wide Web Conference*. 237–247.

[6] Giandomenico Di Domenico, Jason Sit, Alessio Ishizaka, and Daniel Nunan. 2021. Fake news, social media and marketing: A systematic review. *Journal of Business Research* 124 (2021), 329–341.

[7] Yingtong Dou, Kai Shu, Congying Xia, Philip S Yu, and Lichao Sun. 2021. User preference-aware fake news detection. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2051–2055.

[8] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[9] Bing He, Mustaque Ahamad, and Srijan Kumar. 2021. Petgen: Personalized text generation attack on deep sequence embedding-based classification models. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 575–584.

[10] Benjamin D Horne, Jeppe Nørregaard, and Sibel Adali. 2019. Robust fake news detection over time and attack. *ACM Transactions on Intelligent Systems and Technology (TIST)* 11, 1 (2019), 1–23.

[11] Thomas N Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

[12] Thai Le, Suhang Wang, and Dongwon Lee. 2020. Malcom: Generating malicious comments to attack neural fake news detection models. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 282–291.

[13] Angsheng Li and Yicheng Pan. 2016. Structural information and dynamical complexity of networks. *IEEE Transactions on Information Theory* 62, 6 (2016), 3290–3339.

[14] Angsheng Li, Xianchen Yin, and Yicheng Pan. 2016. Three-dimensional gene map of cancer cell types: Structural entropy minimisation principle for defining tumour subtypes. *Scientific Reports* 6 (2016), 1–26.

[15] Angsheng Li, Xianchen Yin, Bingxiang Xu, Danyang Wang, Jimin Han, Yi Wei, Yun Deng, Ying Xiong, and Zhihua Zhang. 2018. Decoding topologically associating domains with ultra-low resolution Hi-C data by graph structural entropy. *Nature Communications* 9 (2018), 1–12.

[16] Jintang Li, Tao Xie, Liang Chen, Fenfang Xie, Xiangnan He, and Zibin Zheng. 2021. Adversarial attack on large scale graph. *IEEE Transactions on Knowledge and Data Engineering* 35, 1 (2021), 82–95.

[17] Yiwei Liu, Jiamou Liu, Zijian Zhang, Liehuang Zhu, and Angsheng Li. 2019. REM: From structural entropy to community structure deception. *Advances in Neural Information Processing Systems* 32 (2019).

[18] Yi Ju Lu and Cheng Te Li. 2020. GCAN: Graph-aware co-attention networks for explainable fake news detection on social media. In *58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*. Association for Computational Linguistics (ACL), 505–514.

[19] Yuefei Lyu, Xiaoyu Yang, Jiaxin Liu, Sihong Xie, Philip Yu, and Xi Zhang. 2023. Interpretable and effective reinforcement learning for attacking against graph-based rumor detection. In *2023 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–9.

[20] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks. In *IJCAI International Joint Conference on Artificial Intelligence*, Vol. 2016. 3818–3824.

[21] Van-Hoang Nguyen, Kazunari Sugiyama, Preslav Nakov, and Min-Yen Kan. 2020. Fang: Leveraging social context for fake news detection using graph representation. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 1165–1174.

[22] Diogo Pacheco, Pik-Mai Hui, Christopher Torres-Lugo, Bao Tran Truong, Alessandro Flammini, and Filippo Menczer. 2021. Uncovering coordinated networks on social media: methods and case studies. In *Proceedings of the international AAAI conference on web and social media*, Vol. 15. 455–466.

[23] Yicheng Pan, Feng Zheng, and Bingchen Fan. 2021. An Information-theoretic Perspective of Hierarchical Clustering. *arXiv preprint arXiv:2108.06036* (2021).

[24] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[25] Yuxiang Ren, Bo Wang, Jiawei Zhang, and Yi Chang. 2020. Adversarial active learning based heterogeneous graph neural network for fake news detection. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 452–461.

[26] Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. Csi: A hybrid deep model for fake news detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 797–806.

[27] Yu Shang, Yudong Zhang, Jiansheng Chen, Depeng Jin, and Yong Li. 2023. Transferable Structure-based Adversarial Attack of Heterogeneous Graph Neural Network. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 2188–2197.

[28] Claude E Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal* 27, 3 (1948), 379–423.

[29] Chengcheng Shao, Giovanni Luca Ciampaglia, Onur Varol, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. 2018. The spread of low-credibility content by social bots. *Nature communications* 9, 1 (2018), 1–9.

[30] Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. 2019. defend: Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 395–405.

[31] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. 2020. Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big data* 8, 3 (2020), 171–188.

[32] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter* 19, 1 (2017), 22–36.

[33] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

[34] Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science* 359, 6380 (2018), 1146–1151.

[35] Haoran Wang, Yingtong Dou, Canyu Chen, Lichao Sun, Philip S Yu, and Kai Shu. 2023. Attacking Fake News Detectors via Manipulating News Social Engagement. In *Proceedings of the ACM Web Conference 2023*. 3978–3986.

[36] Marcin Waniek, Tomasz P Michalak, Michael J Wooldridge, and Talal Rahwan. 2018. Hiding individuals and communities in a social network. *Nature Human Behaviour* 2, 2 (2018), 139–147.

[37] Junran Wu, Xueyuan Chen, Ke Xu, and Shangzhe Li. 2022. Structural entropy guided graph hierarchical pooling. In *ICML*. PMLR, 24017–24030.

[38] Xianghua Zeng, Hao Peng, and Angsheng Li. 2023. Effective and Stable Role-Based Multi-Agent Collaboration by Structural Information Principles. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 10 (Jun. 2023), 11772–11780. doi:10.1609/aaai.v37i10.26390

[39] Xianghua Zeng, Hao Peng, Angsheng Li, Chunyang Liu, Lifang He, and Philip S Yu. 2023. Hierarchical State Abstraction Based on Structural Information Principles. In *IJCAI*.

[40] Mengmei Zhang, Xiao Wang, Chuan Shi, Lingjuan Lyu, Tianchi Yang, and Junping Du. 2023. Minimum topology attacks for graph neural networks. In *Proceedings of the ACM Web Conference 2023*. 630–640.

[41] Peican Zhu, Zechen Pan, Yang Liu, Jiwei Tian, Keke Tang, and Zhen Wang. 2024. A general black-box adversarial attack on graph-based fake news detectors. *arXiv preprint arXiv:2404.15744* (2024).

[42] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. 2018. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2847–2856.

A Framework Details

A.1 Primary Notations

Table 6: Notation Glossary

Notation	Description
$G = (V, E)$	The undirected graph with vertex set and edge set.
v, e	The single vertex and undirected edge.
H, T	The structural entropy and encoding tree.
T	The one- or high-layer encoding tree.
λ, ν, α	The root node, leaf nodes, and other nodes.
$g_\alpha, \mathcal{V}_\alpha$	The terms associated with the vertex subset V_α.
K	The maximal height of the encoding trees.
$\mathcal{I}, c$	The influence metric and adjusting parameter.
x, q	The random variable and probability density function.
U, P	The sets of users and posts.
u, p	The single user account and single news post.
m, n	The respective numbers of users and posts.
X, Y	The feature matrix and label set.
h, y	The hidden representation and single label.
f_θ	The GNN prediction function parameterized by θ.
σ, Δ	The sigmoid function and attack budget.
$\mathcal{N}$	The set of multiple agents.
$\mathcal{S}, \mathcal{A}$	The state and action spaces.
$\mathcal{P}, \mathcal{R}$	The transition and reward functions.
s, a, r	The single state, action, and reward.
$\pi, \mathcal{Q}$	The policy network and value function.

A.2 Limitations

In our study, we focus on attacking graph-based news detectors by manipulating the network structure and classify malicious accounts based on their structural characteristics and the influence they exert on the target post. We draw on prior research [32, 35] to define three types of malicious accounts—bots, cyborgs, and crowd workers. These account types represent varying levels of influence within the network, with bots exerting low influence, cyborgs having medium influence, and crowd workers having high influence. This classification is rooted in the idea that different account types interact with the network in distinct ways, and understanding their influence helps us design more targeted and effective attack strategies. However, we acknowledge that malicious behaviors also include other actions such as spam, scams, and social engineering attacks. These forms of maliciousness are important but are outside the scope of our current work, which focuses specifically on structural manipulation of the network. In future research, we plan to expand our framework to include content-based features to capture a broader range of malicious accounts, including those engaged in spam or scam activities. By incorporating these features, we aim to improve the generalizability of SI2AF and extend it to cover a wider variety of malicious behaviors.

On the other hand, we primarily simulate network dynamics around malicious accounts during the attack process to model certain dynamic aspects. However, we have not yet accounted for the dynamic behaviors of genuine users, which would further complicate the network structure during an attack. We plan to extend our framework in future journal versions to handle dynamic network structures. This will include implementing mechanisms for real-time adjustments to attack strategies, allowing the framework to better align with the continuously evolving nature of social networks. By incorporating dynamic changes into our model, we aim to make the framework more applicable to real-world scenarios, where networks are constantly changing.

A.3 Ethical Statement

Our proposed adversarial attack framework targets the evaluation and enhancement of graph-based news detectors' robustness. We explicitly disavow any unethical use of this framework and emphasize that our research is focused on strengthening detection systems for societal benefit. Our work aligns with the broader goal of promoting the integrity and security of AI-driven systems, particularly in the fight against misinformation and online manipulation.

A.4 Time Complexity

We outline SI2AF's attacking process against a fake news detector in Algorithm 3. The proposed SI2AF operates through several sequential stages: first, hierarchical structure identification for the user-post graph $G_{up} = (U, P, E_{up})$, which has a computational cost of $O(|E_{up}| + (|U| + |P|) \cdot \log^2(|U| + |P|))$; second, multiple agent design with a time complexity of $O(|U| \cdot \log |U|)$; and finally, a subgraph attack for each target, where agents coordinate their actions in the joint action space of $|U_m| \cdot |P_\alpha|$, with $|P_\alpha|$ representing the number of posts in the associated subgraph G_α.

Algorithm 3: SI2AF Attacking against Fake News Detector

Input: maximal timesteps t_{max}, update interval t_{up}
Output: agent policies π_b, π_c, π_w

$G_{up} \leftarrow$ construct the user-post graph
$T^*_{up} = \arg\min_{T_{up}} \{H^{T_{up}}(G_{up})\}$
$U_m = U_b \cup U_c \cup U_w \leftarrow$ categorize malicious accounts via Algorithm 1
for $p \in P_t$ **do**
  $E'_{up} \leftarrow E_{up}$
  $G_\alpha = (U, P_\alpha, E^{up}_\alpha) \leftarrow$ derive the associated subgraph
  P^f_α and $P^r_\alpha \leftarrow$ extract the fake and real news in G_α
  while $t < t_{max}$ **do**
    $\boldsymbol{a^b_t} \leftarrow \pi_b(s_t, U_b, G_\alpha)$
    $\boldsymbol{a^c_t} \leftarrow \pi_c(s_t, U_c, G_\alpha)$
    $\boldsymbol{a^w_t} \leftarrow \pi_w(s_t, U_w, G_\alpha)$
    a^b_t, a^c_t, and $a^w_t \leftarrow$ individually sample $\boldsymbol{a^b_t}$, $\boldsymbol{a^c_t}$, and $\boldsymbol{a^w_t}$ via Equation 11
    $a_t = (u_t, p_t) \leftarrow$ sample single-agent actions a^b_t, a^c_t, and a^w_t
    $E'_{up} = E'_{up} \bigcup \{(u_t, p_t)\}$
    if $t \mod t_{up} == 0$ **then**
      π_b, π_c, and $\pi_w \leftarrow$ update policies by minimizing loss in Equation 13

B Proof of Theorem 4.1

Proof. Considering the transformation:

$$x' = -\frac{x}{b} \cdot \log_2 \left(\frac{c}{b}x\right), \tag{16}$$

which consists of a linear term, $-\frac{x}{b}$, and a logarithmic term, $\log_2(\frac{c}{b}x)$. For $x \in [1, \frac{b}{2}]$ and given the condition $0 < c \leq \frac{2}{e}$, the argument of the logarithm, $\frac{c}{b}x$, is strictly positive. Consequently, both $-\frac{x}{b}$ and $\log_2(\frac{c}{b}x)$ are continuous functions on $x \in [1, \frac{b}{2}]$. Hence, their product, y, is continuous over this interval.

To examine the differentiability of x', we compute the derivative of variable x' with respect to x using standard differentiation rules. Applying the product rule yields:

$$\frac{\mathrm{d}x'}{\mathrm{d}x} = -\frac{1}{b} \cdot \left(\log_2 \frac{c}{b} + \log_2 x + \log_2 e\right). \tag{17}$$

Setting the derivative equal to zero to find critical points:

$$\frac{\mathrm{d}x'}{\mathrm{d}x} = 0, \tag{18}$$

we obtain:

$$\log_2 \frac{c}{b} + \log_2 x + \log_2 e = 0, \tag{19}$$

which simplifies to:

$$x = \frac{b}{ec}. \tag{20}$$

Given the constraints $0 < c \leq \frac{2}{e}$ and the fact that $1 \leq x \leq \frac{b}{2}$, we observe the following relationship:

$$x \leq \frac{b}{2} \leq \frac{b}{ec}. \tag{21}$$

Therefore, within the interval $[1, \frac{b}{2}]$, the variable x' increases monotonically with the variables x, since the derivative $\frac{\mathrm{d}x'}{\mathrm{d}x}$ remains positive, ensuring its monotonic behavior.

Furthermore, the derivative $\frac{\mathrm{d}x'}{\mathrm{d}x}$, as given by Equation 17, is a monotonically decreasing function of x. We now calculate its minimum and maximum values over the interval $x \in [1, \frac{b}{2}]$ as follows:

$$\left(\frac{\mathrm{d}x'}{\mathrm{d}x}\right)_{min} = \left(\frac{\mathrm{d}x'}{\mathrm{d}x}\right)_{x=\frac{b}{2}} = \frac{1}{b} \cdot \log_2 \frac{2}{ec} \geq 0, \tag{22}$$

$$\left(\frac{\mathrm{d}x'}{\mathrm{d}x}\right)_{max} = \left(\frac{\mathrm{d}x'}{\mathrm{d}x}\right)_{x=1} = \frac{1}{b} \cdot \log_2 \frac{b}{ec} \geq 0. \tag{23}$$

To derive the probability density function $q_1(x')$, we user the relationship between the probability density functions of x and x':

$$q_1(x') = q_0(x) \left|\frac{\mathrm{d}x}{\mathrm{d}x'}\right|. \tag{24}$$

Since $\frac{\mathrm{d}x'}{\mathrm{d}x} \geq 0$ over $x \in [1, \frac{b}{2}]$, we have:

$$0 \leq q_0(x)_{min} \cdot \left|\frac{\mathrm{d}x}{\mathrm{d}x'}\right|_{min} \leq q_1(y) \leq q_0(x)_{max} \cdot \left|\frac{\mathrm{d}x}{\mathrm{d}x'}\right|_{max}. \tag{25}$$

Substituting the bounds for $q_0(x)$ and $\left|\frac{\mathrm{d}x'}{\mathrm{d}x}\right|$, we obtain the inequality:

$$q_1(x') \leq \frac{b}{1 - \log_2 ec}. \tag{26}$$

This completes the proof. □

C Evaluation Details

C.1 Datasets

The statistics of benchmark datasets, Politifact and Gossipcop, are summarized in Table 7.

Table 7: Dataset statistics.

Datasets	Nodes	Users	Posts	Edges	Targets
Politifact	276,858	276,277	581	1,074,890	62
Gossipcop	575,993	565,660	10,333	3,084,931	1547

C.2 Detectors.

We train all GNN-based detectors, GAT, GCN, SAGE, Bi-GCN, and GCAN, to optimize their balanced and accurate detection capabilities for fake and real news posts, using metrics accuracy and F1-score, as detailed in Table 8.

Table 8: GNN-based detection performance.

Detection Model	Politifact Dataset		Gossipcop Dataset	
	Accuracy	F1	Accuracy	F1
GCN	0.8157	0.8024	0.9383	0.9348
GAT	0.8354	0.8340	0.9316	0.9266
GraphSAGE	0.8108	0.8102	0.9252	0.9206
GCAN	0.8475	0.8465	0.9142	0.9081
Bi-GCN	0.8084	0.8052	0.8916	0.8840

D Framework Scalability

To validate the scalability of our framework for large social networks, we have developed a strategy to partition the network into local subgraphs based on vertex connectivity. This approach facilitates parallel processing of multiple local subgraphs, which reduces overall time complexity and improves scalability. We perform detailed analyses of training and inference times (ms) using a larger-scale Weibo dataset [20], as shown in the table 9. The results demonstrate that, despite the additional overhead from hierarchical community identification and subgraph targeting, the time costs introduced by our framework remain comparable to those of the original attack algorithms. Even for large social networks, the parallelized processing ensures that computational overhead stays within acceptable limits.

Table 9: Time analysis of our SI2AF and MARL baseline in the Weibo dataset.

Method	Training Time	Inference Time
MARL	537.29	32.65
SI2AF	542.35	36.17

Table 10: Performance Comparison among single-agent variants of MARL and SI2AF within the Politifact and Gossipcop datasets.

Agent	Method	Politifact Fake News					Politifact Real News				
		GAT	GCN	SAGE	Bi-GCN	GCAN	GAT	GCN	SAGE	Bi-GCN	GCAN
Bot	MARL	0.34 ± 0.01	**0.22** ± 0.01	0.36 ± 0.03	**0.22** ± 0.04	0.26 ± 0.05	0.16 ± 0.01	0.50 ± 0.01	**0.32** ± 0.08	0.36 ± 0.04	0.07 ± 0.01
	SI2AF	**0.37** ± 0.03	0.20 ± 0.01	**0.39** ± 0.04	**0.22** ± 0.03	**0.31** ± 0.01	**0.35** ± 0.07	**0.51** ± 0.05	0.26 ± 0.02	**0.41** ± 0.01	**0.15** ± 0.03
Cyborg	MARL	**0.33** ± 0.02	**0.22** ± 0.01	0.33 ± 0.01	**0.12** ± 0.01	0.16 ± 0.02	0.16 ± 0.01	**0.50** ± 0.02	**0.39** ± 0.02	0.27 ± 0.03	0.10 ± 0.01
	SI2AF	0.20 ± 0.03	0.19 ± 0.01	**0.38** ± 0.03	0.10 ± 0.01	**0.23** ± 0.02	**0.47** ± 0.15	**0.50** ± 0.04	0.29 ± 0.02	**0.33** ± 0.02	**0.17** ± 0.02
Worker	MARL	**0.34** ± 0.02	**0.21** ± 0.01	0.33 ± 0.01	0.12 ± 0.02	0.16 ± 0.01	0.16 ± 0.01	0.48 ± 0.02	0.32 ± 0.08	0.27 ± 0.01	0.10 ± 0.01
	SI2AF	0.31 ± 0.05	0.20 ± 0.06	**0.37** ± 0.02	**0.21** ± 0.01	**0.30** ± 0.01	**0.29** ± 0.03	**0.49** ± 0.04	**0.35** ± 0.08	**0.43** ± 0.02	**0.16** ± 0.03
Agent	**Method**	**Gossipcop Fake News**					**Gossipcop Real News**				
		GAT	**GCN**	**SAGE**	**Bi-GCN**	**GCAN**	**GAT**	**GCN**	**SAGE**	**Bi-GCN**	**GCAN**
Bot	MARL	**0.82** ± 0.02	0.29 ± 0.03	0.14 ± 0.01	0.22 ± 0.02	**0.73** ± 0.04	0.18 ± 0.05	0.40 ± 0.10	0.32 ± 0.01	**0.43** ± 0.03	**0.4** ± 0.01
	SI2AF	0.73 ± 0.02	**0.36** ± 0.02	**0.19** ± 0.02	**0.5** ± 0.01	0.49 ± 0.02	**0.30** ± 0.01	**0.47** ± 0.01	**0.34** ± 0.01	0.41 ± 0.01	0.39 ± 0.03
Cyborg	MARL	**0.81** ± 0.01	0.68 ± 0.02	**0.13** ± 0.03	0.08 ± 0.01	0.69 ± 0.06	0.23 ± 0.03	**0.45** ± 0.01	0.32 ± 0.01	**0.42** ± 0.05	**0.41** ± 0.02
	SI2AF	0.71 ± 0.06	**0.81** ± 0.03	0.10 ± 0.01	**0.53** ± 0.03	**0.70** ± 0.02	**0.27** ± 0.04	0.37 ± 0.02	**0.37** ± 0.03	0.35 ± 0.02	0.21 ± 0.09
Worker	MARL	**0.82** ± 0.02	0.48 ± 0.03	0.13 ± 0.02	0.22 ± 0.01	0.71 ± 0.03	**0.27** ± 0.03	0.45 ± 0.05	0.31 ± 0.01	0.12 ± 0.02	0.30 ± 0.02
	SI2AF	0.76 ± 0.03	**0.54** ± 0.02	**0.16** ± 0.01	**0.51** ± 0.03	**0.73** ± 0.03	0.14 ± 0.02	**0.48** ± 0.04	**0.32** ± 0.09	**0.39** ± 0.03	**0.37** ± 0.02

E Single-agent Variant Comparison

We compare the single-agent variants of our SI2AF framework with the multi-agent baseline, MARL [35]. Each variant controls a distinct group of malicious accounts (bots, cyborgs, or crowd workers) to execute attacks aimed at misclassifying fake and real news. Table 10 summarizes the average attack success rates and standard deviations across all five GNN-based detectors.

In the Bot and Worker variants of SI2AF, we observed that SI2AF outperforms MARL in nearly 80% of attack scenarios, demonstrating a clear performance advantage. However, in the Cyborg variant, the performance difference between SI2AF and MARL is less pronounced. This variation can be attributed to two key factors: the classification of malicious accounts and the subgraph attack strategy employed by SI2AF.

The MARL baseline relies on local degree features for account classification, assuming that accounts with higher degrees have more influence. However, due to the heavy-tailed degree distribution in social networks, this approach results in a relatively fixed distribution of Worker accounts, which tend to have higher degrees. Consequently, Worker accounts are less adaptable and struggle to effectively attack target accounts with fewer structural connections, leading to weaker attack performance. In contrast, SI2AF uses global structural information for account classification, resulting in a more balanced distribution of Worker accounts and, consequently, better attack performance.

In the Bot variant, SI2AF's subgraph attack strategy, including indirect and feedback attacks, is particularly effective. These strategies work well for Bot accounts, which have larger budgets and stronger internal collaboration, providing more flexibility in influencing the network. This is why the performance difference between SI2AF and MARL is more pronounced in the Bot variant.

To clarify, although the performance of SI2AF varies across agent types (Bots, Cyborgs, and Workers), the agents do not operate on entirely disjoint sets of predictions. Instead, each agent performs optimally based on its specific characteristics (such as account influence and budget). However, this does not imply that each agent works with completely separate sets of predictions; rather, each agent can influence predictions in different ways, depending on its unique influence within the network.