Topology-Aware Gated Graph Neural Network for Social Bot Detection

Pi Jiebin[1,2], Yuxin Huang[1,2], Yan Xiang[1,2],
Ran Song[1,2], Yantuan Xian[1,2*], Zhengtao Yu[1,2]
[1]Faculty of Information Engineering,
Kunming University of Science and Technology, Kunming, China
[2]Yunnan Key Laboratory of Artificial Intelligence, Kunming, China
pijiebin@stu.kust.edu.cn, song_ransr@163.com, huangyuxin2004@163.com
{ yanx, xianyt, yuzt}@kust.edu.cn

Abstract

The rapid growth of social networks has led to a surge in social bots, which often disseminate low-quality content and may manipulate public opinion, posing threats to online security. Although recent GNN-based bot detection methods perform strongly, they still face two major challenges. First, deep GNNs are prone to over-smoothing: neighbor aggregation blends bot and human node representations, obscuring bot-specific features. Second, social graphs are dominated by human–human and human–bot connections, while direct bot–bot links are scarce, making it difficult for effective bot representations to propagate within GNNs. To address these issues, we propose a Topology-Aware Gated Graph Neural Network (TopGateGNN) to detect social bots. TopGateGNN employs topology-aware data augmentation to synthesize realistic bot nodes that preserve the original graph structure, mitigating class imbalance; it also introduces a hierarchical gating mechanism that restructures node embeddings into a tree format, selectively filtering noise and enhancing discriminative features. Experiments on three standard benchmark datasets show that TopGateGNNconsistently surpasses leading baselines in highly imbalanced settings, delivering superior accuracy and robustness.

1 Introduction

The swift progress in social media has escalated the proliferation of social bots, which engage in automated information manipulation, posing significant threats to cybersecurity (Ferrara et al., 2016; Höhne et al., 2025). Leveraged by malicious entities, social bots disseminate misinformation and generate fabricated interactions, thereby eroding the integrity of information ecosystems and public trust (Cresci, 2020; Orabi et al., 2020; Xu et al., 2020; Cresci et al., 2023; Starbird, 2019; Zannettou et al., 2019). Therefore, effective social bot detection is crucial to ensure cybersecurity. Many methods are proposed to effectively detect social media bots, necessitating robust detection methods to safeguard cybersecurity (Hagen et al., 2022). To address the above issues, many methods have been proposed to effectively detect social media bots.

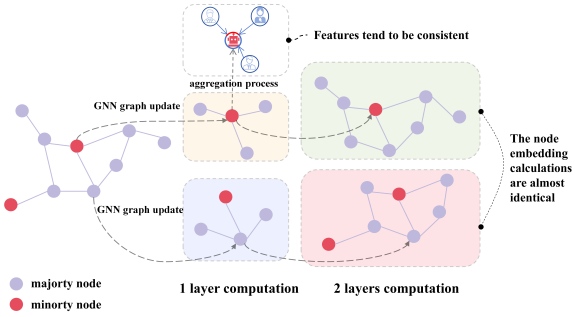


Figure 1: The figure shows the class imbalance and over-smoothing problems faced by GNNs in social bot detection, showing how minority class features become indistinguishable during multi-layer message aggregation, and how node embeddings become similar as the number of layers increases.

Current social bot detection approaches include feature-based, text-based, and graph-based methodologies. Feature-based methods (Kudugunta and Ferrara, 2018; Yang et al., 2020) depend on manually engineered features, which are susceptible to class imbalance and ineffective at detecting sparse bot signals (Feng et al., 2022b). Text-based approaches (Dukić et al., 2020; Heidari and Jones, 2020; Guo et al., 2021; Nedungadi et al., 2025) leverage natural language processing (NLP) to derive features, but produce confounded representations from sophisticated bot-generated text, exacerbating over-smoothing challenges (Cresci, 2020). In contrast, graph-based methodologies, particularly those that utilize graph neural networks (GNN), such as graph convolutional net-

*Corresponding author.

Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 235–245

works (GCN) (Kipf and Welling, 2016; Mendoza et al., 2020; Yang et al., 2024), efficiently capture the structural properties of social networks.

Graph Neural Networks (GNNs) effectively detect social bots by capturing network structures and interactions (Dehghan et al., 2023). Deep GNNs, by modeling multi-hop relationships, can better uncover features of complex bot behaviors (Li et al., 2020). However, significant challenges remain. As shown in Figure 1. Primarily, class imbalance in datasets causes GNNs to favor majority class features, suppressing minority class features and reducing detection accuracy (Feng et al., 2022b). Moreover, node connections in social networks are predominantly human-to-human, diluting bot-specific features. Additionally, traditional Graph Convolutional Networks (GCNs) exacerbate over-smoothing through neighbor aggregation, blurring distinctions between bot and human accounts (Kipf and Welling, 2016; Song et al., 2023). Recent data augmentation (Shi et al., 2024) and aggregation methods (Song et al., 2023) attempt to address these issues but often overlook graph topology or fail to effectively counter minority class feature suppression. These issues make bot detection difficult.

To address these challenges, we introduce TopGateGNN , a framework designed to simultaneously mitigate class imbalance and over-smoothing. Our approach comprises two key components. Initially, we propose a topology-aware data augmentation strategy that dynamically generates synthetic samples near minority class nodes. our topology-aware sampling method leverages local topological properties to create high-quality samples. These samples preserve network connectivity while strengthening the connection between bot and human users.Additionally, we introduce a hierarchical recursive gating mechanism that transforms node embeddings into tree-structured representations and iteratively captures multi-hop neighborhood information. By dynamically weighting and filtering neighbor information based on attribute and structural heterogeneity, this mechanism preserves essential structural and attribute features while mitigating over-smoothing. Our key contributions are as follows:

- We propose a Topology-Aware Sampling method that generates high-quality synthetic samples in regions with sparse minority classes, enhancing bot feature representations in human-dominated networks.
- We introduce a hierarchical recursive gating mechanism that transforms embeddings into tree-structured representations, capturing multi-hop patterns and retaining discriminative features to mitigate over-smoothing.
- Evaluations on three benchmark datasets demonstrate the superior performance and robustness of TopGateGNN across various GNN models, validating its effectiveness in addressing class imbalance and over-smoothing.

2 Related work

2.1 Twitter Bot Detection

Early social bot detection relied on hand-crafted features from tweet content (Cresci, 2020) and user metadata (Yang et al., 2020; Cai et al., 2017), constrained by static patterns. Deep learning methods (e.g., (Kudugunta and Ferrara, 2018; Wei and Nguyen, 2019; Stanton and Irissappane, 2019)) utilized multidimensional data to improve performance. However, feature-based methods are challenged by class imbalance and sparse signals, while deep learning approaches suffer from over-smoothing due to feature confounding.

2.2 Class Imbalance in Bot Detection

Class imbalance in social networks impedes bot detection by underrepresenting minority classes. Prior work (Zhao et al., 2024) shows that synthetic minority nodes improve classification, but aligning them with graph topology remains challenging. OS-GNN (Shi et al., 2024) applies SMOTE to feature and neighborhood spaces but overlooks edge structure, weakening local connectivity. GraphSMOTE (Zhao et al., 2021) preserves neighborhoods but struggles with complex topologies. SMOTENN (Kudugunta and Ferrara, 2018) generates disconnected nodes, limiting pattern capture. These methods fail to prevent feature dilution in human-centric networks. TopGateGNN generates topology-aware bots, preserving connectivity to address imbalance.

2.3 Graph-based Bot Detection

Recent advances in bot detection utilize graph-based methods to leverage social network relations. Graph neural networks (GNNs), like GAT (Veličković et al., 2017), aggregate neighborhood

information via message passing. GCNN (Ali Alhosseini et al., 2019) introduced GNNs to social user classification, leveraging node and edge structures for detection. BotRGCN (Feng et al., 2021b) applies Relational GCNs (Schlichtkrull et al., 2018) with multimodal data to detect disguised bots and coordinated attacks. BotMoE (Liu et al., 2023) combines metadata, text, and network features using a community-aware mixture-of-experts model to enhance detection. Yang et al. (Yang et al., 2023) proposed a heterogeneous graph modeling user relations and text, improving detection accuracy. Despite progress, over-smoothing and limited generalization remain challenges (Ma et al., 2021; Song et al., 2023).

3 Methodology

3.1 Overview

Problem Definition. We model a social network as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, where $\mathcal{V} = \{v_i\}_{i=1}^{n}$ represents user accounts, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes interaction edges (e.g., follows, mentions), and $\mathbf{X} \in \mathbb{R}^{n \times d}$ contains node feature vectors $\mathbf{x}_i \in \mathbb{R}^d$ (e.g., post frequency, profile attributes). Each node $v_i \in \mathcal{V}$ has a binary label $y_i \in \{0, 1\}$, with $y_i = 1$ for social bots and $y_i = 0$ for human accounts. Graph neural networks (GNNs) update node embeddings via message passing:

$$h_v^{(k)} = \text{AGG}\left(h_v^{(k-1)}, \{h_u^{(k-1)} \mid u \in \mathcal{N}(v)\}\right), \tag{1}$$

where $\mathcal{N}(v)$ is the neighbor set of node v, and $\text{AGG}(\cdot)$ is an aggregation function (e.g., mean, sum). Class imbalance and over-smoothing hinder social bot detection: imbalance biases GNNs toward the majority class, while deep layers cause embeddings to converge (over-smoothing), reducing discriminability. The overall framework is shown in Figure 2

3.2 Topology-Aware Sampling

3.2.1 Topology-Aware Weighting

To integrate graph topology into synthetic node generation, we assign weights to minority-class nodes based on degree centrality and local clustering coefficient. Specifically, nodes with high centrality and low clustering are prioritized to better capture the structure of sparse subregions. The weight w_i for each minority class node i is defined as:

$$w_i = \frac{\deg(i)/(c_i + \delta)}{\sum_{j \in \mathcal{M}} \deg(j)/(c_j + \delta)}, \tag{2}$$

where $\deg(i)$ denotes the degree of node i, and c_i represents the clustering coefficient of node i, calculated as the ratio of actual to possible edges in the subgraph induced by node i and its neighbors. $\mathcal{M}$ denotes the set of minority class nodes in the training set, and δ as a small positive constant ensuring numerical stability.

3.2.2 Topology-Aware Node Generation

Based on the computed topological weights, we propose a Topology-Aware Sampling (TAS) strategy that adaptively allocates the number of synthetic nodes to improve class balance. For each minority-class node i, the number of synthetic samples is set to $n_i = \lfloor w_i \cdot N \rfloor$, where N is the total number of synthetic samples determined by the imbalance ratio.

Each synthetic node is generated by interpolating between a minority node $\mathbf{x}_i$ and one of its nearest neighbors $\mathbf{x}_j$ in the feature space:

$$\mathbf{x}_{\text{new}} = \mathbf{x}_i + \lambda(\mathbf{x}_j - \mathbf{x}_i) + \epsilon, \tag{3}$$

where $\lambda \in [0, 1]$ is a random interpolation coefficient, and $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is Gaussian noise added for diversity, with σ as a small constant.

To ensure quality, synthetic nodes are filtered based on their proximity to original samples and their connectivity to high-weight nodes.

3.2.3 Topology-Constrained Edge Generation

To effectively incorporate synthetic nodes into the graph, we propose an edge generation mechanism that integrates feature similarity with topological constraints, ensuring that the augmented graph reflects realistic connectivity patterns. For each synthetic node $\mathbf{x}_{\text{new}}$, we apply the K-nearest neighbors (KNN) algorithm to identify the top K most similar existing nodes in the feature space.

An edge is established between $\mathbf{x}_{\text{new}}$ and a candidate node $\mathbf{x}_j$ if the following condition holds:

$$\text{dist}(\mathbf{x}_{\text{new}}, \mathbf{x}_j) < \theta \quad \text{and} \quad \mathbf{x}_j \in \mathcal{N}_K(\mathbf{x}_{\text{new}}), \tag{4}$$

where $\text{dist}(\cdot)$ denotes Euclidean distance, θ is a distance threshold, and $\mathcal{N}_K(\cdot)$ denotes the set of K nearest neighbors.

To preserve topological fidelity, we further constrain connections by requiring that the candidate node $\mathbf{x}_j$ has degree greater than a threshold ($\deg(j) > \delta_d$) and clustering coefficient below another threshold ($c_j < \delta_c$). This encourages connections to structurally informative but sparsely connected nodes, aligning with the topology of minority class regions.

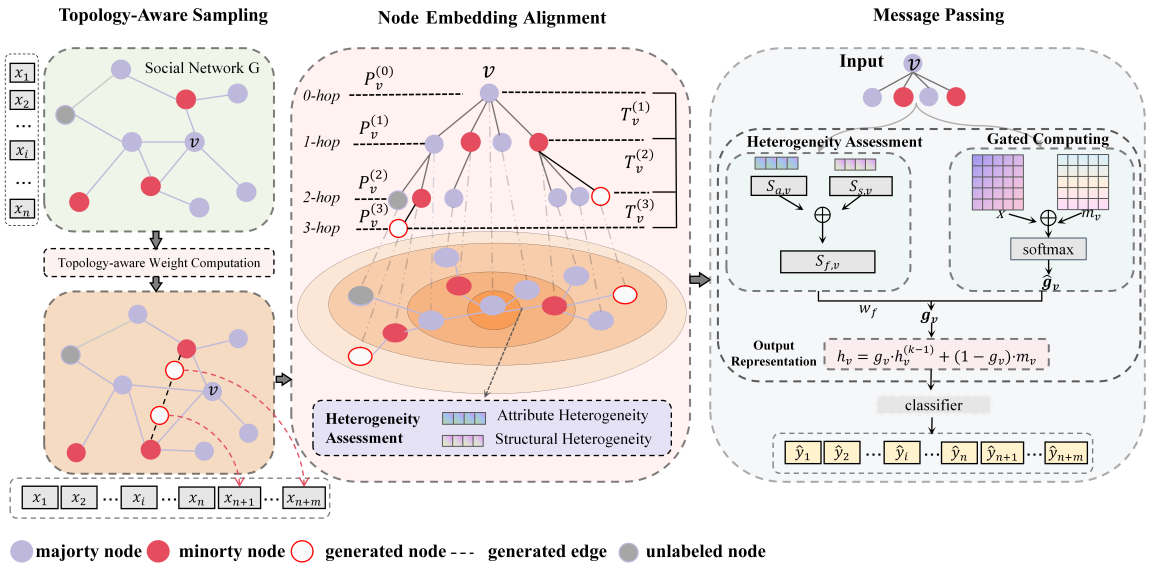


Figure 2: Overview of the TopGateGNN framework. For each node v, a rooted k-hop tree T_v is constructed to capture hierarchical neighborhood information. The embedding $h_v \in \mathbb{R}^D$ is partitioned into $k+1$ segments using learned split points $P_v^{(0)}, \ldots, P_v^{(k)}$, assigning disjoint dimensions to l-hop neighbors. Attribute heterogeneity $s_{\alpha,v}$ and structural heterogeneity $s_{s,v}$ are computed and fused as $s_{f,v}$, guiding gated message passing where embedding $h_v^{(k)}$ is updated using previous layer embedding $h_v^{(k-1)}$, neighbor messages $m_v^{(k)}$, and gating vector $g_v^{(k)}$. Final embeddings are used for prediction $\hat{y}_v$.

3.3 Node Embedding Alignment

3.3.1 Rooted-Tree Hierarchical Mapping

We represent each node v's neighborhood as a k-hop rooted tree $\mathcal{T}_v$, with v as the root (see Figure 2 for an illustration). This tree organizes neighbors into layers based on their hop distance from v, enabling the isolation of multi-hop features. To map the hierarchical structure of $\mathcal{T}_v$ to the node embedding $\mathbf{h}_v \in \mathbb{R}^D$ while preserving multi-hop information, we partition the embedding into $k+1$ segments using split points $0 = P_v^{(0)} \leq P_v^{(1)} \leq \cdots \leq P_v^{(k)} = D$, where $P_v^{(l)}$ marks the boundary for encoding features of l-hop neighbors, preventing feature mixing across hops to mitigate over-smoothing, as illustrated in Figure 2. These split points are dynamically predicted via a learnable gating mechanism detailed in Section 3.3.3.

3.3.2 Heterogeneity Assessment

To capture the complex connectivity patterns in social networks, we introduce a heterogeneity assessment mechanism that adaptively adjusts message-passing weights by evaluating differences in node features and local structures. This mechanism comprises two components: attribute heterogeneity and structural heterogeneity.

Attribute heterogeneity $s_{a,v}$ quantifies feature variance among a node's neighbors and is normalized via a sigmoid function:

$$s_{a,v} = \sigma\left(\frac{1}{|\mathcal{N}(v)|}\sum_{u\in\mathcal{N}(v)}(\mathbf{h}_u^{(k-1)} - \bar{\mathbf{h}}_v^{(k-1)})^2\right), \tag{5}$$

where $\mathbf{h}_u^{(k-1)}$ is the embedding of neighbor node u at layer $k-1$, and $\bar{\mathbf{h}}_v^{(k-1)}$ is the mean neighbor embedding. The set $\mathcal{N}(v)$ denotes the neighbors of node v, and $\sigma(\cdot)$ is the sigmoid function.

Structural heterogeneity $s_{s,v}$ captures local topological variation and is computed using the entropy of the neighbor degree distribution:

$$s_{s,v} = \sigma\left(-\sum_b p_b \log(p_b + \varepsilon)\right), \tag{6}$$

where p_b is the proportion of neighbor degrees falling into bin b, and ε is a small constant to ensure numerical stability.

The final heterogeneity score $s_{f,v}$ is obtained by fusing the two components through a learnable transformation:

$$s_{f,v} = \sigma\left(\mathbf{W}_f[s_{a,v}; s_{s,v}]\right), \tag{7}$$

where $\mathbf{W}_f$ is a trainable weight matrix, and $[\cdot;\cdot]$ denotes vector concatenation.

3.3.3 Gated Embedding Refinement

To address heterogeneity in social networks, we introduce a dynamic heterogeneity gating mechanism, where the initial gating vector is computed via linear projection:

$$\hat{g}_v^{(k)} = \text{cumsoft}\left(\mathbf{W}^{(k)}[h_v^{(k-1)}; m_v^{(k)}]\right), \quad (8)$$

where $h_v^{(k-1)}$ denotes the embedding from the previous layer, $m_v^{(k)} = \frac{1}{|\mathcal{N}(v)|}\sum_{u\in\mathcal{N}(v)} h_u^{(k-1)}$ represents the aggregated neighbor message, $\mathbf{W}^{(k)}$ denotes a learnable weight matrix, and $\text{cumsoft}(\cdot) = \text{cumsum}(\text{softmax}(\cdot))$ denotes the cumulative softmax operation, ensuring a monotonically increasing gating signal for stable embedding updates. To further ensure monotonicity across layers, the gating vector is refined by

$$\tilde{g}_v^{(k)\text{mono}} = \tilde{g}_v^{(k-1)} + (1 - \tilde{g}_v^{(k-1)}) \circ \hat{g}_v^{(k)}, \quad (9)$$

where $\tilde{g}_v^{(k-1)}$ is the previous layer's refined gating vector and $\circ$ denotes element-wise multiplication. The heterogeneity score $s_{f,v}$ further refines the gating vector as

$$\tilde{g}_v^{(k)} = \sigma\left(\tilde{g}_v^{(k)\text{mono}} + w_f s_{f,v}\right), \quad (10)$$

where w_f is a trainable parameter. Finally, the node embedding is updated through gated fusion:

$$h_v^{(k)} = \tilde{g}_v^{(k)} \circ h_v^{(k-1)} + (1 - \tilde{g}_v^{(k)}) \circ m_v^{(k)}, \quad (11)$$

3.4 Training

We train the TopGateGNN for social bot detection in a supervised manner, where each node v is labeled $y_v \in \{0, 1\}$, with $y_v = 0$ indicating a human account and $y_v = 1$ indicating a bot. The training objective is to minimize a weighted binary cross-entropy loss:

$$\mathcal{L} = -\frac{1}{N}\sum_{v=1}^{N} w_v[y_v \log(\hat{y}_v) + (1-y_v)\log(1-\hat{y}_v)], \quad (12)$$

where N denotes the total number of nodes, $\hat{y}_v = \sigma(W_{\text{out}} h_v^{(K)} + b_{\text{out}})$ represents the predicted probability of a node being a bot, derived from the final gated embedding $h_v^{(K)} \in \mathbb{R}^D$ through a linear layer. To address class imbalance, we employ class-balanced weights defined as $w_v = \frac{N}{2N_{y_v}}$, where N_{y_v} represents the number of nodes in class y_v. Detailed description of these trainings is provided in Appendix A.1.

4 Experimental Setups

Datasets. We use three social bot detection benchmark datasets to evaluate the performance of TopGateGNN : **Cresci-15** (Cresci et al., 2015), **TwiBot-20** (Feng et al., 2021a), and **MGTAB** (Shi et al., 2023). Detailed descriptions of these datasets are provided in Appendix A.2.

Baselines. We compare it against a diverse set of baseline methods for social bot detection, categorized into three groups: Graph Neural Network (GNN)-based approaches (e.g., BotRGCN (Feng et al., 2021b), SimpleHGN (Lv et al., 2021), SeBot (Yang et al., 2024), and RGT (Feng et al., 2022a)); sample synthesis-based approaches (e.g., GraphSmote (Zhao et al., 2021) and OS-GNN (Shi et al., 2024)); and feature-based approaches (e.g., BotBuster (Ng and Carley, 2023), BotMoe (Liu et al., 2023), RoBERTa (Liu et al., 2019), and SGBot (Yang et al., 2020)). Detailed descriptions of these baselines are provided in Appendix A.3.

Implementation. Experiments were conducted on two NVIDIA RTX 3090 GPUs using PyTorch, PyTorch Geometric, and scikit-learn. Models were trained using Adam, with GNNs set to 4 layers and a 256-dimensional hidden layer. Hyperparameters included a learning rate of 0.001, dropout of 0.3, and weight decay of 1e-5, with nearest neighbor $k = 5$ and $\lambda = 0.8$ for preprocessing. Models were trained until convergence for up to 300 epochs.

5 Results and Analysis

5.1 Main Results

We evaluate TopGateGNN on nine representative baselines from three widely used Twitter bot detection benchmarks: Cresci-15, TwiBot-20, and MGTAB. As shown in Table 1.

Existing GNN-based methods offer limited gains over traditional approaches on MGTAB, with only marginal F1 improvements. In contrast, TopGateGNN consistently outperforms all baselines across Cresci-15, TwiBot-20, and MGTAB by leveraging topology-aware augmentation to alleviate class imbalance. It surpasses OS-GNN and Simple-HGN on MGTAB in F1 and balanced accuracy, respectively, and outperforms SEBot in balanced accuracy on TwiBot-20. On Cresci-15, TopGateGNN achieves substantial performance gains. These results confirm that TopGateGNN effectively addresses class imbalance and over-smoothing through topology-aware augmentation, dynamic gating, and heterogeneity-

Method	Cresci-15			TwiBot-20			MGTAB		
	F1-score	**Accuracy**	**bAcc**	**F1-score**	**Accuracy**	**bAcc**	**F1-score**	**Accuracy**	**bAcc**
GraphSmote	96.26	96.56	95.98	71.50	76.40	75.25	79.21	83.28	81.96
SGBot	77.91	77.10	80.40	84.15	79.50	81.00	-	-	-
BotBuster	97.53	96.90	96.28	82.47	79.34	78.30	-	-	-
RoBERTa	97.22	96.41	95.84	76.58	71.93	70.59	-	-	-
Simple-HGN	94.68	93.13	87.65	86.33	83.93	83.28	79.43	89.08	87.68
BotRGCN	96.50	97.60	87.00	87.25	85.75	86.50	79.83	89.37	84.20
RGT	97.78	96.63	96.07	87.80	86.36	86.70	79.97	89.47	83.56
BotMoE	**98.82**	98.50	87.50	88.04	86.59	86.55	79.83	88.97	84.10
OS-GNN	96.38	96.65	96.35	82.30	82.49	82.41	85.39	86.75	87.18
SEBot	-	-	-	**88.41**	86.85	87.31	81.61	90.24	84.38
TopGateGNN (Ours)	98.55	**98.62**	**98.60**	88.12	**87.25**	**87.50**	**87.31**	**91.62**	**89.32**

Table 1: Performance comparison of various methods on Cresci-15, TwiBot-20, and MGTAB datasets (F1-score, Accuracy, and Balanced Accuracy). Bold indicates the best, underline indicates the second best. '-' indicates that the baseline is not scalable to the corresponding dataset.

aware modeling, enhancing the detection of bot-specific multi-hop anomalies.

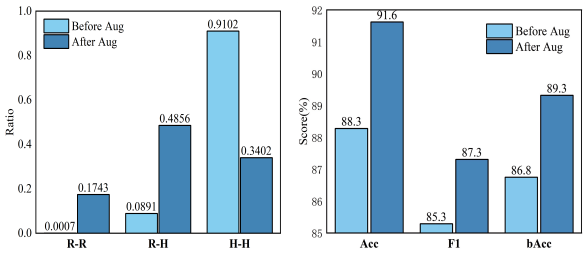
5.2 Augmentation Study

To assess the effectiveness of topology-aware augmentation in enhancing robot account connectivity, we analyzed edge type proportions (robot-robot [R-R], robot-human [R-H], human-human [H-H]) on the MGTAB dataset before and after augmentation, as depicted in Figure 3. Initially, R-R edges were scarce (0.07%), with H-H edges dominating (91.02%) and R-H edges at 8.91%, reflecting sparse robot connections. Post-augmentation, R-R edges increased to 17.43%, R-H edges to 48.56%, and H-H edges decreased to 34.02%, indicating enhanced robot connectivity and strengthened bot-specific patterns.

New edges connect these synthetic social bot nodes to existing nodes based on feature proximity and topological constraints. Since the new nodes are robots, all new edges are R-R or R-H edges, with R-H edges dominating because there are more human nodes in the original graph. This enhancement enhances the connectivity of robots, makes robot-specific patterns more prominent, enhances the model's capture of minority class node features.

5.3 Gating Analysis

To investigate the role of the proposed gating mechanism in TopGateGNN for social bot detection, we visualize the distribution of gating signals across six graph convolutional layers, separated by user (Label 0) and bot (Label 1) labels. As shown in Figure 4, in the first layer, user nodes exhibit a broader signal distribution with lower intensity, while bot



(a) Edge Ratio (b) Performance

Figure 3: The proportion and performance changes of different edge types before and after enhancement on MGTAB.

nodes display a more concentrated and stronger signal. With deeper layers, bot signals remain focused and prominent, whereas user signals become increasingly varied, reflecting adaptive information flow control.

This divergence in signal dynamics effectively mitigates over-smoothing in deep GNNs by limiting excessive feature mixing for user nodes while enhancing targeted aggregation for bot nodes. Consequently, TopGateGNN maintains distinct representation boundaries between users and bots, improving classification accuracy and interpretability in social bot detection across layers.

5.4 Over-Smoothing Analysis

We evaluated its performance on the MGTAB dataset across varying layer depths (2, 3, 4, 6, 8, 12), comparing it with baselines GCN (Kipf and Welling, 2016), BotRGCN (Feng et al., 2021b), and OS-GNN (Shi et al., 2024). We also conducted an ablation study by removing the gating mechanism

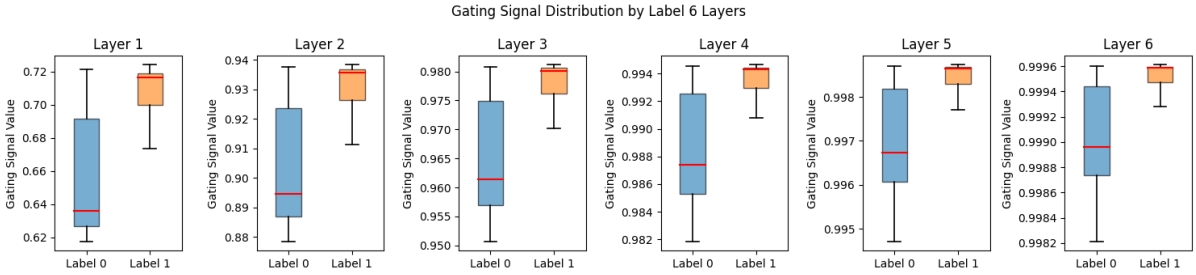


Figure 4: Visualization of gating signals.

(Ours w/o Gating) to examine its role.

As shown in Figure 5, TopGateGNN maintains stable performance across all layers, consistently outperforming baselines. In contrast, GCN, BotRGCN, and OS-GNN exhibit a marked performance decline as layers deepen, reflecting their susceptibility to over-smoothing and loss of discriminative power between bots and users. The ablation study further reveals that without the gating mechanism, performance degrades similarly to baselines, underscoring its critical role in preserving feature distinctiveness.

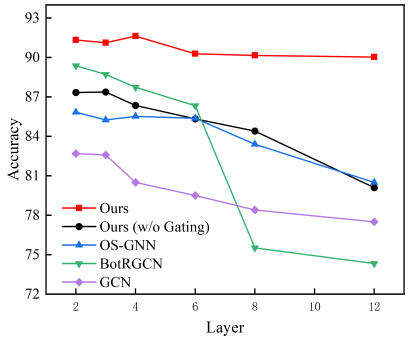


Figure 5: Test accuracy across increasing layers.

5.5 Topology-Aware Sampling Study

To evaluate our topology-aware sampling (TAS), we conducted experiments on the MGTAB dataset, comparing TAS with fixed sampling ratios. Unlike fixed ratios, TAS dynamically adjusts the sampling ratio based on minority class node topology, generating synthetic samples that mitigate class imbalance.

As shown in Figure 6, TAS outperforms fixed sampling ratios across all metrics. Fixed sampling ratios peak then decline in performance, as excessive sampling introduces noise and insufficient sampling underrepresents the minority class. In contrast, TAS optimizes sampling ratios, improving the representational capacity of graph neural networks in imbalanced networks. By integrating topological information, TAS generates high-quality samples to alleviate class imbalance and refine classification boundaries, thereby improving TopGateGNN 's pattern detection capabilities for specific robots.

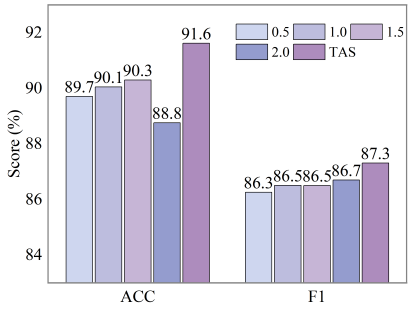


Figure 6: Performance comparison at different sampling rates on the MGTAB dataset.

Table 2: Data Augmentation Effects with Different K Values on MGTAB

K	Dist.	Acc	F1	bAcc
1	0.7845	90.40	87.66	87.61
3	0.8082	90.24	87.68	88.24
5	**0.8231**	**91.62**	87.31	**89.32**
7	0.8323	89.95	87.24	88.02

5.6 Synthetic Study

To evaluate topology-aware data augmentation for social bot detection in TopGateGNN on imbalanced networks, we conducted experiments on the MGTAB dataset, analyzing how the number of nearest neighbors ($K \in \{1, 3, 5, 7\}$) affects performance. We assessed two metrics: (1) Dist., measuring sample diversity; and (2) performance metrics, for detection capability. Results are shown in Table 2.

As K increases, Dist. rises from 0.7845 to 0.8323, as larger K includes diverse neighbors, improving minority class representation. At K=5, TopGateGNN achieves optimal performance with a Dist. of 0.8231, balancing diversity and deviation.

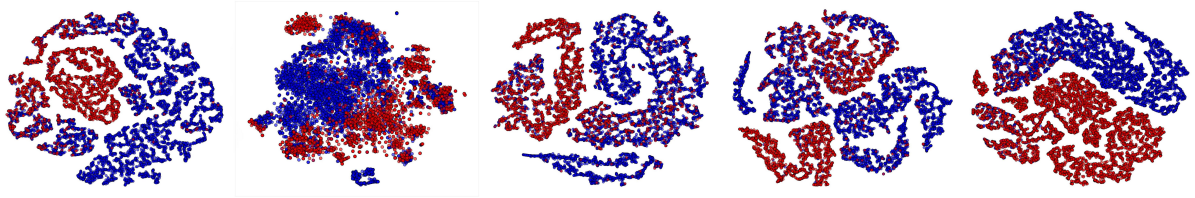


Figure 7: Node embedding visualization (bots are shown in red, while humans are shown in blue).

At K=7, performance declines due to excessively divergent samples impairing generalization.

5.7 Ablation Study

Ablation experiments on MGTAB assess the contribution of each TopGateGNN component, using Bare GNN as the backbone. See Table 3 for results.

- Without a heterogeneous gating mechanism, the model overmixes features during aggregation, making bot and human nodes hard to distinguish in deeper layers due to over-smoothing.
- Without gating, the model cannot separate multi-hop information, leading to mixed embeddings that weaken its ability to detect complex bot behaviors.
- Without synthesizing minority class nodes, GNNs rely on the original imbalanced graph, where human connections dominate, diluting robot-specific features.

Model	Acc	F1	bAcc
Full TopGateGNN	91.62	87.31	89.32
w/o HeteGating	$90.61_{\downarrow 1.01}$	$86.87_{\downarrow 0.44}$	$88.34_{\downarrow 0.98}$
w/o Gating	$89.37_{\downarrow 2.25}$	$86.07_{\downarrow 1.24}$	$88.12_{\downarrow 1.20}$
w/o TAS	$88.28_{\downarrow 3.34}$	$85.29_{\downarrow 2.02}$	$86.75_{\downarrow 2.57}$

Table 3: Ablation study.

5.8 Visualization Analysis

We visualize node embeddings from various methods with t-SNE(Van der Maaten and Hinton, 2008) to compare their performance in social bot detection. As shown in figure 7, our model achieves better clustering and class separation compared to baseline methods.

BotRGCN forms compact clusters, but class boundaries are blurred, with notable overlap. OS-GNN has overlapping clusters and poor class separation. GCN clusters with intra-cluster compactness lacking and unclear class boundaries. GAT slightly improves local structure capture, but its distribution is scattered, with poor class transitions. In contrast, our model shows tighter clusters and clearer class boundaries, with compact intra-cluster distributions and distinct inter-cluster separation. These results suggest that our model captures node similarities and ensures smoother class transitions globally, improving the quality of node representations and class differentiation accuracy.

6 Conclusion

We propose TopGateGNN , which employs topology-aware sampling to generate synthetic minority class samples and uses a hierarchical gating mechanism to create tree-like node embeddings, addressing class imbalance while reducing over-smoothing and improving feature discrimination. Experiments on three benchmark datasets show TopGateGNN surpasses existing methods. These results confirm TopGateGNN 's effectiveness in social bot detection, offering avenues for graph-based anomaly detection research.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Nos. 62266028, 62266027, 62466029), the Key Projects of Basic Research in Yun- nan Province (Nos. 202301AS070047, 202501AS070147), the Major Science and Technology Projects of Yunnan Province(202402AD080002, 202402AG050007). The authors would like to thank anonymous reviewers for their comments.

Limitations

TopGateGNN effectively relieves class imbalance and over-smoothing in social bot detection but has limitations. Evaluations on Cresci-15, TwiBot-20, and MGTAB datasets may not reflect evolving bot behaviors, as (Cresci et al., 2023) note issues with dataset labeling and collection. Testing on real-time networks could strengthen validation, but

shifting interactions complicate this. Our topology-aware sampling relies on local topological properties, which may overlook global or temporal network structures, as explored by (Yang et al., 2024). Although large language models could extract user features (Yang and Menczer, 2023), we prioritized graph neural networks, leaving LLM integration for future work. These limitations suggest directions for improving TopGateGNN 's applicability in diverse networks.

References

Seyed Ali Alhosseini, Raad Bin Tareaf, Pejman Najafi, and Christoph Meinel. 2019. Detect me if you can: Spam bot detection using inductive representation learning. In *Companion proceedings of the 2019 world wide web conference*, pages 148–153.

Chiyu Cai, Linjing Li, and Daniel Zeng. 2017. Detecting social bots by jointly modeling deep behavior and content information. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1995–1998.

Stefano Cresci. 2020. A decade of social bot detection. *Communications of the ACM*, 63(10):72–83.

Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. 2015. Fame for sale: Efficient detection of fake twitter followers. *Decision Support Systems*, 80:56–71.

Stefano Cresci, Kai-Cheng Yang, Angelo Spognardi, Roberto Di Pietro, Filippo Menczer, and Marinella Petrocchi. 2023. Demystifying misconceptions in social bots research. *arXiv preprint arXiv:2303.17251*.

Ashkan Dehghan, Kinga Siuta, Agata Skorupka, Akshat Dubey, Andrei Betlen, David Miller, Wei Xu, Bogumił Kamiński, and Paweł Prałat. 2023. Detecting bots in social-networks using node and structural embeddings. *Journal of Big Data*, 10(1):119.

David Dukić, Dominik Keča, and Dominik Stipić. 2020. Are you human? detecting bots on twitter using bert. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 631–636. IEEE.

Shangbin Feng, Zhaoxuan Tan, Rui Li, and Minnan Luo. 2022a. Heterogeneity-aware twitter bot detection with relational graph transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3977–3985.

Shangbin Feng, Zhaoxuan Tan, Herun Wan, Ningnan Wang, Zilong Chen, Binchi Zhang, Qinghua Zheng, Wenqian Zhang, Zhenyu Lei, Shujie Yang, et al. 2022b. Twibot-22: Towards graph-based twitter bot detection. *Advances in Neural Information Processing Systems*, 35:35254–35269.

Shangbin Feng, Herun Wan, Ningnan Wang, Jundong Li, and Minnan Luo. 2021a. Twibot-20: A comprehensive twitter bot detection benchmark. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 4485–4494.

Shangbin Feng, Herun Wan, Ningnan Wang, and Minnan Luo. 2021b. Botrgcn: Twitter bot detection with relational graph convolutional networks. In *Proceedings of the 2021 IEEE/ACM international conference on advances in social networks analysis and mining*, pages 236–239.

Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. 2016. The rise of social bots. *Communications of the ACM*, 59(7):96–104.

Qinglang Guo, Haiyong Xie, Yangyang Li, Wen Ma, and Chao Zhang. 2021. Social bots detection via fusing bert and graph convolutional networks. *Symmetry*, 14(1):30.

Loni Hagen, Stephen Neely, Thomas E. Keller, Ryan Scharf, and Fatima Espinoza Vasquez. 2022. Rise of the machines? examining the influence of social bots on a political discussion network. *Social Science Computer Review*, 40(2):264–287.

Maryam Heidari and James H Jones. 2020. Using bert to extract topic-independent sentiment features for social media bot detection. In *2020 11th IEEE annual ubiquitous computing, electronics & mobile communication conference (UEMCON)*, pages 0542–0547. IEEE.

Jan Karem Höhne, Joshua Claassen, Saijal Shahania, and David Broneske. 2025. Bots in web survey interviews: A showcase. *International Journal of Market Research*, 67(1):3–12.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Sneha Kudugunta and Emilio Ferrara. 2018. Deep neural networks for bot detection. *Information Sciences*, 467:312–322.

Guohao Li, Chenxin Xiong, Ali Thabet, and Bernard Ghanem. 2020. Deepergcn: All you need to train deeper gcns. *arXiv preprint arXiv:2006.07739*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yuhan Liu, Zhaoxuan Tan, Heng Wang, Shangbin Feng, Qinghua Zheng, and Minnan Luo. 2023. Botmoe: Twitter bot detection with community-aware mixtures of modal-specific experts. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 485–495.

Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang. 2021. Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 1150–1160.

Jiaqi Ma, Junwei Deng, and Qiaozhu Mei. 2021. Subgroup generalization and fairness of graph neural networks. *Advances in Neural Information Processing Systems*, 34:1048–1061.

Marcelo Mendoza, Maurizio Tesconi, and Stefano Cresci. 2020. Bots in social and interaction networks: detection and impact estimation. *ACM Transactions on Information Systems (TOIS)*, 39(1):1–32.

Prema Nedungadi, G. Veena, Kai-Yu Tang, Remya R. K. Menon, and Raghu Raman. 2025. Ai techniques and applications for online social networks and media: Insights from bertopic modeling. *IEEE Access*, 13:37389–37407.

Lynnette Hui Xian Ng and Kathleen M Carley. 2023. Botbuster: Multi-platform bot detection using a mixture of experts. In *Proceedings of the international AAAI conference on web and social media*, volume 17, pages 686–697.

Mariam Orabi, Djedjiga Mouheb, Zaher Al Aghbari, and Ibrahim Kamel. 2020. Detection of bots in social media: a systematic review. *Information Processing & Management*, 57(4):102250.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*, pages 593–607. Springer.

Shuhao Shi, Kai Qiao, Chen Chen, Jie Yang, Jian Chen, and Bin Yan. 2024. Over-sampling strategy in feature space for graphs based class-imbalanced bot detection. In *Companion Proceedings of the ACM Web Conference 2024*, pages 738–741.

Shuhao Shi, Kai Qiao, Jian Chen, Shuai Yang, Jie Yang, Baojie Song, Linyuan Wang, and Bin Yan. 2023. Mgtab: A multi-relational graph-based twitter account detection benchmark. *arXiv preprint arXiv:2301.01123*.

Yunchong Song, Chenghu Zhou, Xinbing Wang, and Zhouhan Lin. 2023. Ordered gnn: Ordering message passing to deal with heterophily and over-smoothing. *arXiv preprint arXiv:2302.01524*.

Gray Stanton and Athirai A Irissappane. 2019. Gans for semi-supervised opinion spam detection. *arXiv preprint arXiv:1903.08289*.

Kate Starbird. 2019. Disinformation's spread: bots, trolls and all of us. *Nature*, 571(7766):449–450.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Feng Wei and Uyen Trang Nguyen. 2019. Twitter bot detection using bidirectional long short-term memory neural networks and word embeddings. In *2019 First IEEE International conference on trust, privacy and security in intelligent systems and applications (TPS-ISA)*, pages 101–109. IEEE.

Qiongkai Xu, Lizhen Qu, Zeyu Gao, and Gholamreza Haffari. 2020. Personal information leakage detection in conversations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6567–6580.

Kai-Cheng Yang and Filippo Menczer. 2023. Anatomy of an ai-powered malicious social botnet. *arXiv preprint arXiv:2307.16336*.

Kai-Cheng Yang, Onur Varol, Pik-Mai Hui, and Filippo Menczer. 2020. Scalable and generalizable social bot detection through data selection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 1096–1103.

Yingguang Yang, Qi Wu, Buyun He, Hao Peng, Renyu Yang, Zhifeng Hao, and Yong Liao. 2024. Sebot: Structural entropy guided multi-view contrastive learning for social bot detection. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3841–3852.

Yingguang Yang, Renyu Yang, Yangyang Li, Kai Cui, Zhiqin Yang, Yue Wang, Jie Xu, and Haiyong Xie. 2023. Rosgas: Adaptive social bot detection with reinforced self-supervised gnn architecture search. *ACM Transactions on the Web*, 17(3):1–31.

Savvas Zannettou, Tristan Caulfield, Emiliano De Cristofaro, Michael Sirivianos, Gianluca Stringhini, and Jeremy Blackburn. 2019. Disinformation warfare: Understanding state-sponsored trolls on twitter and their influence on the web. In *Companion proceedings of the 2019 world wide web conference*, pages 218–226.

Tianxiang Zhao, Xiang Zhang, and Suhang Wang. 2021. Graphsmote: Imbalanced node classification on graphs with graph neural networks. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 833–841.

Tianxiang Zhao, Xiang Zhang, and Suhang Wang. 2024. Imbalanced node classification with synthetic oversampling. *IEEE Transactions on Knowledge and Data Engineering*, 36(12):8515–8528.

A Experimental Details

A.1 Training Setting

We optimize $\mathcal{L}$ using the Adam optimizer, with inputs comprising an augmented graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ generated through topology-aware sampling, along with node features $\mathbf{x}_v \in \mathbb{R}^d$. This approach incorporates multi-relational edge support to generate high-quality embeddings, thereby improving social bot detection performance.

A.2 Datasets

Cresci-15 contains 5,301 Twitter accounts, of which 3,351 are bot accounts and 1,950 are human accounts, covering follow and follow relationships. TwiBot-20 contains 11,826 Twitter accounts, of which 6,589 are bot accounts and 5,237 are human accounts, involving follow and follow interactions. MGTAB contains 10,384 accounts, of which 2,830 are bot accounts and 7,554 are human accounts, supporting multiple relationship types. The experiments follow the training, validation, and testing splits defined in (Shi et al., 2023).

A.3 Baselines

To validate our approach, we compare our model with a variety of baseline methods:

- GraphSmote (Zhao et al., 2021) constructed an embedding space to encode the similarities between nodes. New samples are synthesized in this space to ensure their authenticity.
- SGBot (Yang et al., 2020) constructed a dataset A and a rigorous validation system was designed in addition to traditional cross-validation to optimize the performance of the model on the prediction dataset.
- BotBuster (Ng and Carley, 2023) proposed a social robot detection method based on the Mixture of Experts architecture, aiming to address the limitations of existing methods in the face of data loss and cross-platform applications.
- RGT (Feng et al., 2022a) proposed a heterogeneity-aware bot detector that dynamically incorporates and exploits heterogeneous relationships and influence patterns among users.
- SimpleHGN (Lv et al., 2021) proposed a simple and powerful baseline model, SimpleHGN, to promote HGNN research towards a more robust and reproducible development.
- BotRGCN (Feng et al., 2021b) constructed a heterogeneous graph and applied the Relational GCN (RGCN) for detection.
- OS-GNN (Shi et al., 2024) improved the model's ability to recognize minority classes by synthesizing minority class samples in the feature space, avoiding the generation of new graph structures or edges.
- RoBERTa(Liu et al., 2019) is a pre-trained language model based on the Transformer architecture, optimized the BERT pre-training process.
- BotMoe (Liu et al., 2023) introduced a multimodal Twitter robot detection framework with a community-aware MoE architecture, which effectively solved problems such as feature manipulation, camouflage, and cross-community generalization.
- SeBot (Yang et al., 2024) is a social robot detection framework based on graph neural networks, which uses structural entropy to guide multi-view contrastive learning.