# TopoMSG: A Topology-Aware Multi-Scale Graph Network for Social Bot Detection

**Anonymous ACL submission**

## Abstract

Social bots are automated programs designed to spread rumors and misinformation, posing significant threats to the security of the network. Graph Neural Network (GNN)-based social bot detection models are limited by the over-smoothing and over-squashing problems of the message-passing mechanism, making it difficult to effectively extract key high-dimensional topological features and model complex topological structures across different social networks. To address the issue of limited topological feature extraction caused by over-smoothing and over-squashing in GNN-based social bot detection models, we propose a topology-aware multi-scale detection method for social bots. By leveraging local topological layers and a clustering attention mechanism, the approach effectively incorporates topological features into node representations and captures multi-level structural patterns at both global and local scales. Experimental results demonstrate that our model exhibits strong competitiveness on three widely used benchmark datasets, effectively addressing existing methods' limitations in capturing local feature patterns, while also being capable of capturing global features, thereby enhancing the overall modeling of complex structures. We publicly release our code in `https://anonymous.4open.science/r/TopoMSG-2C41/`

## 1 Introduction

Social bots are automated programs operating on social media platforms, where malicious bots significantly threaten network security through mass retweeting, rumor dissemination, election interference, and extremist ideology propagation (Berger and Morgan, 2015; Deb et al., 2019). Social bot detection provides crucial safeguards for maintaining cyberspace security and enhancing platform credibility.

Influenced by a variety of factors including node connectivity patterns, community structures, and user interaction behaviors, the topological architecture of social networks manifests significant complexity and diversity, a characteristic that is especially pronounced in datasets constructed via differing methodologies. Existing social bot detection datasets are primarily constructed based on graph structures. However, the adoption of different sampling methods results in significant variations in the topological features of these datasets. As shown in Figure 1, MGTAB (Shi et al., 2023) is constructed as a scale-free dataset based on topic relevance. TwiBot-20 (Feng et al., 2021) is built as a tree-like structured dataset using breadth-first search. TwiBot-22 (Feng et al., 2022b) is developed as a heterogeneous graph dataset through metadata neighborhood expansion. (Ng and Carley, 2025) shows that bots tend to form star-like network structures, while human users are more likely to exhibit tree-like patterns. The high prevalence of tree-like structures in the TwiBot-20 dataset may pose challenges for bot detection tasks. We further analyzed the proportion of users forming cyclic structures under different relationship types, as presented in Table. 1. The results show that in datasets constructed using hierarchical sampling methods (e.g., TwiBot-20 and TwiBot-22), social bots tend to form cyclic structures in the network, mimicking the interaction

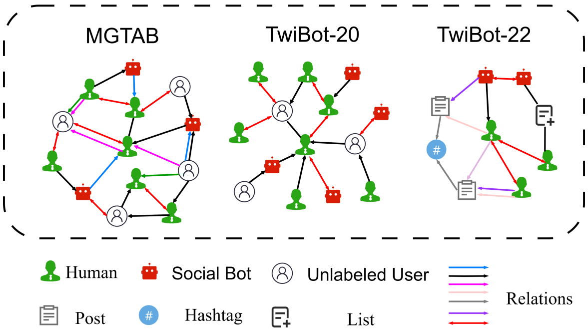


Figure 1: Topological structure features across datasets, Relations represent relationships between different entities, such as followers, friend, comments, retweets, etc.

patterns of human users. However, in the topic-based dataset MGTAB, the proportion of social bots forming such cyclic structures is significantly lower. This indicates that social bots struggle to effectively imitate human behavior when engaging in substantive discussions with real users.

| Dataset | Relation | On Cycle | | Off Cycle | |
|---|---|---|---|---|---|
| | | Users | Bots | Users | Bots |
| Twibot-20 | followers | 56.42% | 43.58% | 44.35% | 55.65% |
| | friends | 36.98% | 63.02% | 59.73% | 40.27% |
| | All Relations | 44.28% | 55.72% | 44.28% | 55.72% |
| MGTAB | followers | 83.51% | 16.49% | 55.01% | 44.99% |
| | friends | 83.98% | 16.02% | 52.86% | 47.14% |
| | mention | 97.05% | 2.95% | 69.75% | 30.25% |
| | reply | 98.93% | 1.07% | 83.85% | 16.15% |
| | quoted | 99.60% | 0.40% | 80.83% | 19.17% |
| | URL | – | – | 29.16% | 70.84% |
| | hashtag | – | – | 75.41% | 24.59% |
| | All Relations | 73.06% | 26.94% | 73.06% | 26.94% |
| TwiBot-22 (subgraph) | followers | 88.14% | 11.86% | 85.91% | 14.09% |
| | friends | 90.30% | 9.70% | 91.49% | 8.51% |
| | All Relations | 88.26% | 11.74% | 88.26% | 11.74% |

Table 1: Comparison of Users and Bots Proportions on and off Cycles. TwiBot-22(subgraph) is a connected subgraph extracted from the homogeneous graph of user nodes in the heterogeneous graph TwiBot-22. "–" indicates that no cyclic structures exist under this relation.

Both message-passing-based (Gilmer et al., 2017) and attention-based (Vaswani et al., 2017) detection methods face distinct challenges in practice. Graph Neural Networks (GNNs) leverage message passing to aggregate features from neighboring nodes and capture structural information (Fu et al., 2023; Huang et al., 2024; Liu et al., 2024). However, they are prone to the problems of over-smoothing (Yang et al., 2020; Rusch et al., 2023) and over-squashing (Topping et al., 2021; Di Giovanni et al., 2023), which hinder the model's ability to learn long-range dependencies. Additionally, aggregation functions are typically designed to aggregate node attribute features while largely ignoring the topological structure attributes of networks, such as connected components, cycles, and holes (Bouritsas et al., 2022). In contrast, attention-based Graph Transformers excel at modeling long-range dependencies. However, they suffer from the local-global chaos problem (Wang et al., 2024) during the integration of local and global information, leading to overfitting and over-globalization (Xing et al., 2024), which negatively impacts model generalization.

Therefore, social bot detection faces two key challenges: (a) How to utilize high-dimensional topological features under different relationships to address the incomplete topological structures caused by sampling. (b) How to separate global and local features to alleviate the impact of local-global chaos. To address these issues, we propose a topology-aware multi-scale social bot detection method.

First, for local feature extraction, we introduce Persistent Homology (Edelsbrunner et al., 2008), a method from topological data analysis, to encode structural features of nodes based on different types of edge relationships, thereby capturing rich topological properties. Persistent Homology has shown strong potential in deep learning (Zia et al., 2024), showcasing its effectiveness especially in graph classification tasks (Aktas et al., 2019). Besides, recent research has also reported significant progress in leveraging Persistent Homology for node classification tasks (Immonen et al., 2023). Second, for global feature extraction, we employ a clustered global attention mechanism to alleviate information imbalance caused by over-globalization. Finally, we adopt a global-local collaborative training strategy to automatically adjust the importance weights of global and local features. Our model integrates global attention mechanisms with message-passing mechanisms, achieving collaborative optimization between global and local features, which enhances detection accuracy and robustness.

Our main contributions are summarized as follows:

- This study is the first to apply Persistent Homology to the task of social bot detection, enhancing node representation by capturing high-dimensional topological features under heterogeneous node relationships.

- We propose a clustered global attention mechanism, which effectively reduces the over-reliance of Graph Transformers on global attention and alleviates local-global chaos caused by over-globalization.

- A collaborative training strategy is employed to automatically balance and optimize the learning process of global and local features.

- Experimental results demonstrate that our method achieves performance comparable to state-of-the-art approaches across multiple datasets, showcasing its strong competitiveness and broad application potential.

## 2 Related Work

Social networks are essentially complex graph structures constituted by massive user interactions, excelling in capturing collective behaviors and propagation patterns compared to text analysis methods. The advent of attention mechanisms has significantly advanced graph neural networks (GNNs). Existing graph transformers approaches can be categorized into two main types: message-passing-based and global-attention-based.

### 2.1 Message Passing Neural Networks

Graph Attention Networks (GATs) (Veličković et al., 2017) demonstrate unique advantages in social bot detection. The classic GAT adaptively assigns inter-node weights through attention mechanisms to capture the most relevant interaction patterns. Inspired by this, the Relational Graph Transformer (RGT) (Feng et al., 2022a) introduces multi-relational attention mechanisms that dynamically adjust weights across semantically distinct relationships, enhancing modeling capacity for complex social networks. The Heterogeneous Graph Transformer (HGT) (Wang et al., 2019) further designs specialized attention mechanisms for diverse user types and interaction patterns in social networks through explicit modeling of the heterogeneity of nodes and edges. These message-passing-based GNNs not only enable flexible node aggregation but also capture diverse structural features, achieving state-of-the-art performance in bot detection tasks.

### 2.2 Global Attention Neural Networks

Transformers provide theoretical foundations for GNNs to capture global features through learnable fully-connected attention graphs (Waswani et al., 2017). NodeFormer (Wu et al., 2022) proposes an all-pair message passing paradigm that reduces computational complexity to linear scale via kernelized Gumbel-Softmax operators, enabling efficient signal propagation on large graphs. SGFormer (Wu et al., 2024) introduces a simplified graph transformer architecture that resolves quadratic overhead through single-layer attention modeling. Despite their demonstrated potential in capturing long-range dependencies, global attention mechanisms remain underexplored in social bot detection.

## 3 Preliminaries

This section formally introduces key concepts employed in our work.

**Persistent Homology**: Topological structures are defined as features invariant under continuous deformations (Zia et al., 2024). Persistent homology identifies multi-scale topological signatures by tracking homology group evolution across filtration scales. Different homology orders represent dimensional features: 0-order (connected components), 1-order (cycles), and 2-order (cavities).

For a $k$-dimensional simplicial complex $C$ (composed of simplices: 0-simplices as vertices, 1-simplices as edges, 2-simplices as triangles) with filtration values $a_0 \leq a_1 \leq \cdots \leq a_n$, we define a filtration process via a filtration function $f$ that generates nested complexes:

$$\emptyset \subset C_1 \subset C_2 \subset \cdots \subset C_n \tag{1}$$

This filtration induces birth/death events of topological structures (connected components, cycles, voids). Each structure is associated with a birth-death pair $(a_i, a_j)$, visualized in persistence diagrams where $x$- and $y$-axes represent birth/death times. Diagonal points denote short-lived local features or noise, while off-diagonal points correspond to persistent global structures.

## 4 Methodology

We propose a topology-aware multi-scale detection model. As shown in Figure 2, the TopoMSG framework integrates persistent homology-based topological learning with multi-head global attention through co-training, enabling joint optimization of global-local feature awareness.

### 4.1 Local Topo Relational Graph Transformer

Inspired by Feng et al. (Feng et al., 2022a), we adopted the architecture of RGT and utilized the attention mechanism to learn diverse node representations under each relation. Given user feature vectors $x_i$ and feature matrix $X$, we first apply linear transformation:

$$x^{(0)} = \text{LeakyReLU}(W \cdot x + b) \tag{2}$$

where $W \in \mathbb{R}^{d \times d}$ and $b$ are learnable parameters.

The message-passing mechanism and multi-head attention self-attention mechanisms have demonstrated outstanding performance in the field of

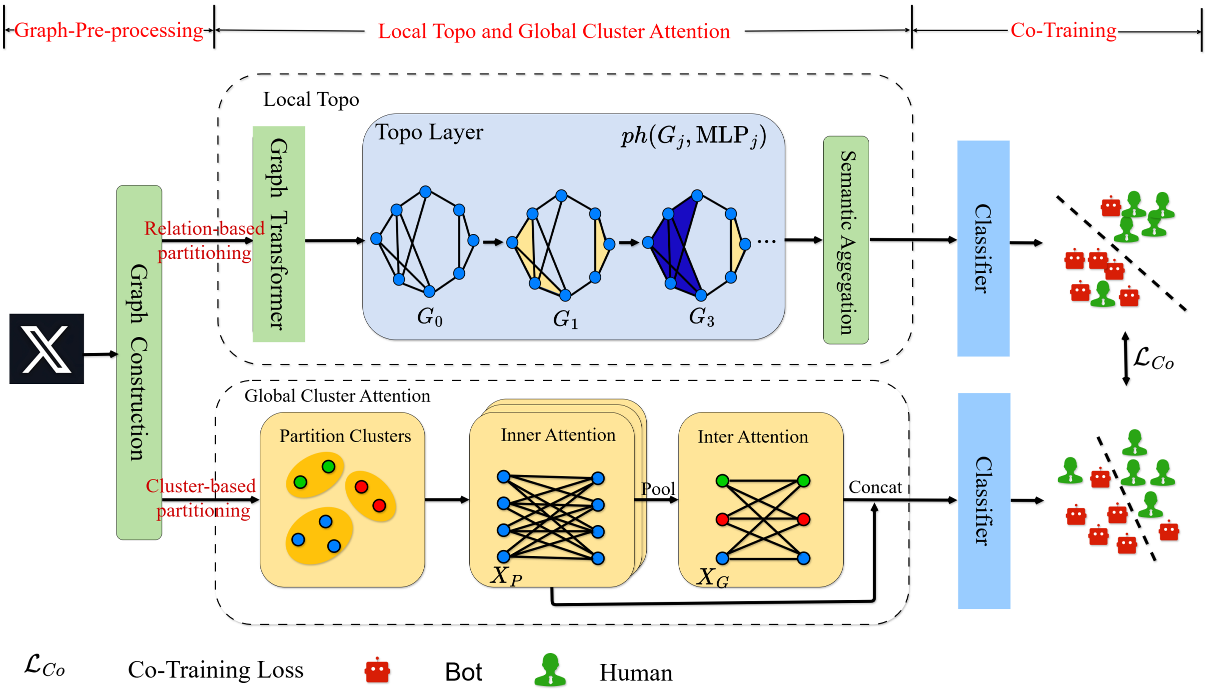


Figure 2: Topology-aware multi-scale Graph Network framework

graph neural networks. Consequently, we employ Graph Transformer layers (GTLayer) to extract shallow topological representations of nodes. The hidden representations $x_j^{(l-1)}$ are fed into Graph Transformer:

$$(q,k,v)_{c,i}^{r(l)} = x_j^{(l-1)} W_{c,(q,k,v)}^{l} + b_{c,(q,k,v)}^{l} \tag{3}$$

where $q$, $k$, $v$ denote query,key,value vectors, respectively; $W_{c,(q,k,v)}^{l}$ and $b_{c,(q,k,v)}^{l}$ are learnable parameters with regard to relation $r$ and head $c$ ; $l$ denotes the l-th layer of Graph Transformer. Then, we leverage the message-passing mechanism to learn the neighborhood unit structure for each node and capture the graph structural information.

$$u_i^{r(l)} = \frac{1}{C}\sum_{c=1}^{C}\sum_{j\in N^r(i)} \alpha_{c,i,j}^{r(l)} v_{c,j}^{r(l)} \tag{4}$$

where $\alpha$ represents the attention scores under different relations and attention heads:

$$\alpha_{c,i,j}^{r}(l) = \frac{\langle q_{c,i}^{r}(l), k_{c,j}^{r}(l)\rangle}{\sum_{u\in N^r(i)}\langle q_{c,i}^{r}(l), k_{c,u}^{r}(l)\rangle} \tag{5}$$

where $\langle\cdot,\cdot\rangle$ denotes exponential-scaled dot-product; $N^r(i)$ represents neighbors under relation $r$.

We employ persistent homology to capture topological features under different relationship types and design the TopoLayer (Horn et al., 2021) to effectively integrate the continuously evolving topological features of nodes into their vector representations. Further details are provided in Section 4.2:

$$u_{Topo,i}^{r(l)} = \mathrm{TopoLayer}(u_i^{r(l)}) \tag{6}$$

At least, using semantic aggregation networks (Feng et al., 2022a) to aggregate node representations across relations while preserving the relation heterogeneity entailed in the social networks:

$$w_d^r = \frac{1}{|V|}\sum_{i\in V} {q_d^{(l)}}^T \cdot \tanh(W_{d,s}^{(l)} \cdot u_{Topo,i}^{r(l)} + b_{d,s}^{(l)}) \tag{7}$$

where $q_d^{(l)}$, $W_{d,s}^{(l)}$, and $b_{d,s}^{(l)}$ are semantic attention parameters. Normalized weights are obtained via:

$$\beta_d^{r(l)} = \frac{\exp(w_d^{r(l)})}{\sum_{k\in R}\exp(w_d^{k(l)})} \tag{8}$$

leading to the final node representation:

$$x_{L,i} = \frac{1}{D}\sum_{d=1}^{D}\left[\sum_{r\in R}\beta_d^{r(l)} \cdot u_{Topo,i}^{r(l)}\right] \tag{9}$$

where $x_{L,i}$ denotes the local topology-aware node embedding.

### 4.2 TopoLayer

In this subsection, we describe the architecture of TopoLayer in detail. We employ persistent homology from topological data analysis to reveal multi-scale hidden shape features (e.g., 0-dimensional connected components, 1-dimensional cycles, 2-dimensional voids). Specifically, we take GNN hidden layer representations as input and maintain identical input-output dimensions, enabling seamless integration with arbitrary GNN layers. First, we map nodes to $k$ distinct real-valued groups through $k$ filtration functions to capture multi-scale topological features. We implement Multilayer Perceptrons (MLPs) as persistent homology filtration functions:

$$u_{i,j}^{r(l)} = f_j(u_i^{r(l)}), \quad j \in (1,k) \tag{10}$$

where $u_{i,j}^{r(l)}$ denotes the filtration value of node $i$, generating $k$ filtration groups. The filtration function $f_j$ is instantiated as an MLP. Each filtration group captures distinct graph attributes. For node filtration values $u_{0,j}^{r(l)}, u_{1,j}^{r(l)}, \ldots, u_{n,j}^{r(l)}$, we filter graph $G$ through nested complexes:

$$G_{i,j} = (V_{i,j}, E_{i,j}), \quad i \in (1,n), j \in (1,k), \tag{11}$$

$$V_j^{r(l)} = \{v \in V \mid v_{i,j}^{r(l)} \leq u_{i,j}^{r(l)}\} \tag{12}$$

$$E_j^i = \{(v,w) \in E \mid \max\{v_{i,j}^{r(l)}, w_{i,j}^{r(l)}\} \leq f_j^i\} \tag{13}$$

This process generates persistence diagrams:

$$\mathrm{ph}(G_j, \mathrm{MLP}_j) = \{D_j^0, D_j^1, \ldots, D_j^m\} \tag{14}$$

where $\mathrm{ph}(G_j, \mathrm{MLP}_j)$ denotes persistence diagrams under filtration function $\mathrm{MLP}_j$, and $D_j^m$ represents diagrams across topological dimensions. We then map persistence diagrams to $d$-dimensional node embeddings via linear projection:

$$t_{i,j}^{r(l)} = \sum_{p=0}^{m} \mathrm{Linear}(D_j^m) \tag{15}$$

Aggregating $k$-scale features with residual connections yields:

$$u_{\mathrm{Topo},i}^{r(l)} = u_{i,j}^{r(l)} + \sum_{j=1}^{k} t_{i,j}^{r(l)} \tag{16}$$

where $u_{\mathrm{Topo},i}^{r(l)}$ denotes topology-enhanced neighborhood features.

### 4.3 Global Cluster Attention

Global-Attention-Based Graph Transformers excel in capturing long-range dependencies, surpassing the capabilities of message-passing mechanisms. However, they suffer from over-globalization, where attention weights disproportionately focus on higher-order neighbors, often neglecting critical information residing in lower-hop neighbors. Our Global Cluster Attention restricts model focus within clusters to address this issue. Using Metis(Karypis and Kumar, 1998), we partition the graph into $m$ non-overlapping clusters and remove edges enabling the nodes to focus more on intra-cluster information.

$$\hat{H}_P^k = \text{Muti-head Attention}\,(X_P) \tag{17}$$

where $X_P$represent the $k$-th cluster node embeddings, and $\hat{H}_P^k$ is the output of multi-head Attention. Then we apply mean pooling to $\hat{H}_P^k$ to obtain cluster representation $P$. Next, we apply mean pooling to $\hat{H}_P^k$in order to obtain a unified representation for clusters:

$$P = [Mean(H_P^k), Mean(H_Q^k), ...] \tag{18}$$

And then, we feed the cluster representations $P$ into MHA:

$$\hat{P} = \text{Muti-head Attention}\,(P) \tag{19}$$

At least, we concat intra-cluster and inter-cluster node representation:

$$X_G = \mathrm{concat}((\hat{H}_P^k, 1^{\frac{N}{P}} \cdot \hat{p}^T)W_f) \tag{20}$$

where $W_f \in R^{2d \times d}$ are learnable parameters and $1^{\frac{N}{P}}$ denotes all-ones vector with a dimensional of $\frac{N}{P}$.

### 4.4 Co-training

We implement co-training to dynamically balance local-global feature importance. Sequentially, we feed the output of Local Topo Graph Transformer $x_{L,i}$ and the Global Cluster Attention $x_{g,i}$ into the classifier to obtain soft labels.:

$$\hat{y}_{l,i} = \mathrm{softmax}\,(\mathrm{Linear}(x_{L,i})) \tag{21}$$

$$\hat{y}_{g,i} = \mathrm{softmax}\,(\mathrm{Linear}(x_{G,i})) \tag{22}$$

Finally, by utilizing the collaborative training loss function, we not only enhance the model's ability

Table 2: Comparison of ACC and F1 scores across different models on multiple datasets. **Bold** values indicate the best results, "-" represents cases where the model failed due to out-of-memory errors.

| Method | Global Attention | Local Attention | MGTAB | | Twibot-20 | | Twibot-22(subgraph) | |
|---|---|---|---|---|---|---|---|---|
| | | | ACC | F1 | ACC | F1 | ACC | F1 |
| GCN | ✗ | ✗ | $81.59 \pm 0.39$ | $60.79 \pm 2.98$ | $73.61 \pm 0.63$ | $75.16 \pm 0.92$ | $90.68 \pm 0.08$ | $52.79 \pm 1.34$ |
| RGCN | ✗ | ✗ | $88.16 \pm 0.61$ | $78.24 \pm 1.13$ | $85.48 \pm 1.31$ | $86.94 \pm 1.92$ | $92.52 \pm 0.17$ | $64.32 \pm 1.60$ |
| GAT | ✗ | ✓ | $78.63 \pm 0.85$ | $43.12 \pm 4.03$ | $77.21 \pm 0.47$ | $79.46 \pm 0.51$ | $91.09 \pm 0.10$ | $55.99 \pm 0.70$ |
| RGT | ✗ | ✓ | $86.80 \pm 0.48$ | $76.02 \pm 1.77$ | $85.53 \pm 1.14$ | $86.81 \pm 1.48$ | $90.05 \pm 0.08$ | $52.27 \pm 2.54$ |
| Nodeformer | ✓ | ✗ | $87.69 \pm 0.20$ | $\mathbf{83.51 \pm 0.50}$ | $86.66 \pm 0.15$ | $88.25 \pm 0.13$ | – | – |
| Ours | ✓ | ✓ | $\mathbf{88.70 \pm 0.27}$ | $79.85 \pm 0.37$ | $\mathbf{87.15 \pm 0.08}$ | $\mathbf{88.64 \pm 0.10}$ | $\mathbf{92.70 \pm 0.09}$ | $\mathbf{66.67 \pm 1.01}$ |

to fit the ground-truth labels but also strengthen the collaborative consistency between the two modules:

$$\mathcal{L}_{co} = \alpha(\mathcal{L}_l + \mathcal{L}_g) + (1-\alpha)(\mathcal{L}_{lg} + \mathcal{L}_{gl}) \quad (23)$$

where $\mathcal{L}_l$ and $\mathcal{L}_g$ denote the losses of the Local Topo module and the Global Cluster Attention module, respectively. $\mathcal{L}_{lg}$ and $\mathcal{L}_{gl}$ represent the losses between the two modules. All loss functions are implemented using the cross-entropy loss formulation. $\alpha$ is the balance factor that controls the relative weights of the individual loss components.

# 5 EXPERIMENTS

## 5.1 Datasets

We selected three publicly available datasets that are widely used in the field of social bot detection. These datasets were constructed differently, resulting in distinct topological structures. MGTAB was constructed under a unified theme, resulting in a relatively dense topological structure. In contrast, TwiBot-20 and TwiBot-22 were sampled based on relational criteria, leading to sparser graph structures. Moreover, since TwiBot-22 is a heterogeneous graph, we extracted only the user-to-user relations, which resulted in a large number of isolated nodes. To ensure graph connectivity, we filtered out these isolated nodes and constructed the TwiBot-22(subgraph). We provide detailed descriptions of the datasets in Appendix. A

## 5.2 Baselines

We compared our model with some classical and state-of-the-art models on graph neural networks to verify the effectiveness of our optimizations and improvements for graph neural networks.

- **Message Passing:** Message Passing Aggregate messages from neighboring nodes to update each node's feature representation. Such as: **GCN**, **RGCN**
- **Local Attention:**Local Attention employ self-attention to compute message or aggregation weight. Such as **RGT**
- **Global Attention:** Global attention mechanisms allow each node to compute its attention weights with respect to all other nodes in the graph. Such as **Nodeformer**

## 5.3 Experimental Setups

In our experimental setup, all models were trained using an RTX 3090 GPU. The learning rate was set to 0.01, dropout to 0.3, and the hidden layer dimension to 128. In our model, we set the topological layer to extract 0-dimensional features.

## 5.4 Main Result

In our experiments, we compared different baseline models by categorizing them based on local attention and global attention mechanisms. The results reported in Table 2 are averaged over five independent runs. From the results , we can draw the following conclusions:

- For graph datasets with complex structures, local information plays a crucial role; in contrast, graphs with simpler structures rely more on global information to capture bot characteristics. Specifically, on the MGTAB dataset, the RGCN model — which is based on message-passing mechanisms and effectively exploits local neighborhood information — achieves an accuracy of 88.16%, outperforming the NodeFormer model that employs global attention mechanisms. However, on the TwiBot series of datasets, where edges are sparsely sampled, message-passing-based models perform suboptimally.
- In imbalanced dataset MGTAB, global features significantly improve recall performance. The NodeFormer model achieves the highest F1-score of 83.51% on the MGTAB dataset,

Table 3: Ablation study. **w/o Global Cluster Attention** refers to a *Local Topo Relational Graph Transformer* module; **w/o Local Topo** denotes the standalone *Global Cluster Attention* module; and **w/o Co-training** represents the variant where global and local features are directly concatenated instead of being trained through the proposed co-training mechanism. Additionally, **RGT** and **Vanilla MHA** serve as the backbone architectures for the Local Topo and Global Cluster Attention modules, respectively.

| Method | MGTAB | | Twibot-20 | | Twibot-22(subgraph) | |
|---|---|---|---|---|---|---|
| | ACC | F1 | ACC | F1 | ACC | F1 |
| Ours | **88.70 ± 0.27** | **79.85 ± 0.37** | **87.15 ± 0.08** | **88.64 ± 0.10** | **92.70 ± 0.09** | **66.67 ± 1.01** |
| w/o Global Cluster Attention | 86.90 ± 0.69 | 76.86 ± 1.40 | 86.61 ± 0.17 | 88.12 ± 0.23 | 90.39 ± 0.09 | 51.83 ± 1.83 |
| w/o Local Topo | 88.41 ± 0.27 | 78.73 ± 0.56 | 86.45 ± 0.35 | 87.98 ± 0.36 | 89.84 ± 0.07 | 50.20 ± 0.67 |
| w/o co-training | 87.71 ± 0.25 | 78.33 ± 0.54 | 85.47 ± 0.12 | 86.74 ± 0.12 | 89.59 ± 0.64 | 37.91 ± 13.15 |
| RGT | 86.80 ± 0.48 | 76.02 ± 1.77 | 85.53 ± 1.14 | 86.81 ± 1.48 | 90.05 ± 0.08 | 52.27 ± 2.54 |
| Vanilla MHA | 85.84 ± 0.98 | 73.32 ± 2.60 | 86.12 ± 0.76 | 87.53 ± 0.84 | 91.00 ± 0.25 | 21.89 ± 6.07 |

which contains 7,451 genuine users and 2,748 bot accounts. This demonstrates the effectiveness of global feature modeling in handling class imbalance.

- Compared to methods focusing on single-scale information processing, our model effectively integrates both local and global information, achieving the best classification accuracy across three different datasets with an average improvement of approximately 0.5 percentage points. This result highlights the strong generalization capability of our model across datasets with varying sampling strategies.

### 5.5 Analysis

In this section, we analyze our model to address the following questions:

- **RQ1:** Does each component of our model contribute significantly to the overall performance?
- **RQ2:** Does the use of heterogeneous graphs help the model capture more topological structural information?
- **RQ3:** Can our model learn more discriminative node representations?

**Ablation Study(RQ1):** We design and conduct a series of ablation experiments to evaluate the contribution of each component in our model. Table 2 summarizes the role of key modules in the overall architecture. Experimental results show that the Local Topo module significantly outperforms its baseline RGT across all three datasets, with an improvement of 1.08% on the TwiBot-20 dataset. The Global Cluster Attention module also surpasses Vanilla MHA by 2.57% on the MGTAB dataset, which has complex local structures, indicating that the clustering mechanism effectively alleviates the over-globalization issue caused by standard global attention. Removing the co-training mechanism and replacing it with direct feature concatenation leads to a substantial performance drop — even falling below the performance of either individual module (as observed on TwiBot-20 and TwiBot-22). This demonstrates that local and global features cannot be effectively fused through simple concatenation, and the proposed co-training strategy enables complementary learning between the two, thereby improving the overall model performance.

Table 4: Performance Metrics for Different Datasets and Relation Types. Heterogeneous refers to the extraction of topological features based on distinct relationship types, whereas Homogeneous denotes the extraction of topological features without differentiating among relationship types.

| Datasets | Relation Type | Acc (%) | F1-score (%) |
|---|---|---|---|
| MGTAB | 1: followers | 85.06 ± 0.57 | 74.09 ± 1.3 |
| | 2: friends | 85.47 ± 1.52 | 74.75 ± 3.07 |
| | 3: mention | 85.12 ± 0.71 | 72.69 ± 1.17 |
| | 4: reply | 84.73 ± 1.04 | 72.33 ± 2.01 |
| | 5: quoted | 85.92 ± 1.00 | 73.86 ± 1.94 |
| | 6: hashtag | 85.00 ± 1.39 | 72.47 ± 2.18 |
| | 7: url | 85.76 ± 0.80 | 74.14 ± 1.67 |
| | Metadata (1+2) | 85.08 ± 0.64 | 73.66 ± 1.31 |
| | Content (3+4+5) | 86.13 ± 0.66 | 75.38 ± 0.94 |
| | Heterogeneous | **86.90 ± 0.69** | **76.86 ± 1.40** |
| | Homogeneous | 86.31 ± 0.33 | 74.74 ± 1.14 |
| TwiBot-20 | 1: followers | 86.56 ± 0.26 | 87.96 ± 0.31 |
| | 2: friends | 86.38 ± 0.14 | 87.84 ± 0.16 |
| | Heterogeneous | **86.61 ± 0.17** | **88.12 ± 0.23** |
| | Homogeneous | 86.38 ± 0.23 | 87.87 ± 0.28 |
| TwiBot-22 (subgraph) | 1: followers | 89.31 ± 0.17 | 44.67 ± 3.28 |
| | 2: friends | 89.65 ± 0.10 | 49.34 ± 2.50 |
| | Heterogeneous | **90.39 ± 0.09** | **51.83 ± 1.83** |
| | Homogeneous | 89.96 ± 0.70 | 50.63 ± 2.49 |

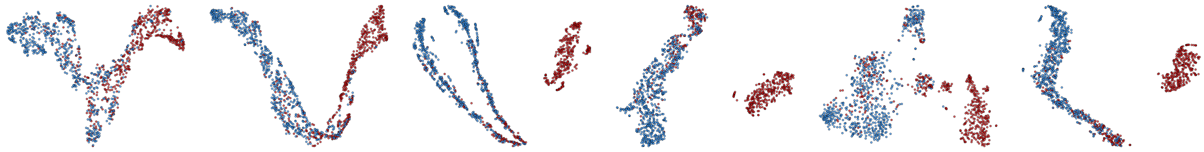
(a) GCN (b) GAT (c) RGCN (d) RGT (e) NodeFormer (f) Ours

Figure 3: TwiBot-20 dataset account node representation visualization. Blue denotes Bot accounts and red denotes Human accounts.

**Relational Topology Analysis(RQ2):** To investigate the impact of topological features from different types of relations in heterogeneous graphs on model performance , we systematically analyzed the experimental results under various relation combinations. As shown in Table 4, We investigated the extraction of topological features from relationally heterogeneous graphs across different types of relations, and compared it with the feature extraction based on homogeneous graphs. Experimental results on three independent datasets show that the method based on relational heterogeneity achieves accuracy rates of 86.90%, 86.61%, and 90.39%, respectively, outperforming the homogeneous graph-based approach by an average of 0.42 percentage points. This demonstrates the effectiveness of modeling relational heterogeneity in our framework. Furthermore, we find that relying solely on a single type of relation for topological feature extraction leads to a significant degradation in model performance. To further evaluate the impact of different relation types, we conducted ablation studies on the MGTAB dataset, where we separately extracted topological features based on user metadata (Metadata) and user content (Content). The results indicate that incorporating content-based topological structures improves both the accuracy (ACC) and F1-Score by more than 1%. This suggests that current social bots still struggle to fully mimic the complex interaction patterns exhibited by genuine users; topological structures can provide valuable discriminative cues for social bot detection.

**Representation Study(RQ3):** To address RQ3, we conducted a two-dimensional visualization of the generated node representations using t-SNE on the TwiBot-20 dataset, as shown in Figure 3. The results show that traditional methods such as GCN and GAT fail to effectively separate different categories of nodes. Although RGCN performs relatively well in node classification, it produces two distinct cluster centers for bot accounts. NodeFormer, while achieving suboptimal accuracy, leads to overly dispersed embeddings for bot accounts. In contrast, both RGT and our model generate more compact embeddings, with our model achieving even tighter clustering, thereby reducing the misclassification rate of genuine users as bots.

## 6 Conclusion

In this paper, we propose a topology-aware multi-scale graph network for social bot detection, aiming to address two key challenges: the difficulty in extracting topological features and the issue of local-global chaos. Specifically, we design both global and local modules to separately model node representations at different scales, thereby mitigating the interference caused by local-global chaos. Meanwhile, we incorporate Persistent Homology, a technique from topological data analysis, into node embeddings to explicitly capture high-order topological information. Extensive experimental results demonstrate that our method significantly improves the accuracy of bot detection across multiple datasets, highlighting the importance of topological features and providing a promising new direction for future research in this field.

## Limitations

## 7 Limitations

**Time Complexity** We employed the Metis tool for hierarchical graph partitioning. However, it introduces notable computational overhead on large-scale datasets. More efficient clustering strategies could improve the model's scalability.

**High-Dimensional Topology** Due to resource and efficiency constraints, we only utilized 0-dimensional persistent homology features (connected components). Higher-dimensional features (e.g., loops and voids) may offer richer structural information. Future work includes exploring efficient ways to incorporate these features to further enhance performance.

## References


Mehmet E Aktas, Esra Akbas, and Ahmed El Fatmaoui. 2019. Persistence homology of networks: methods and applications. *Applied Network Science*, 4(1):1–28.

Jonathon M Berger and Jonathon Morgan. 2015. The isis twitter census: Defining and describing the population of isis supporters on twitter.

Karsten Borgwardt, Elisabetta Ghisu, Felipe Llinares-López, Leslie O'Bray, Bastian Rieck, et al. 2020. Graph kernels: State-of-the-art and future challenges. *Foundations and Trends® in Machine Learning*, 13(5-6):531–712.

Giorgos Bouritsas, Fabrizio Frasca, Stefanos Zafeiriou, and Michael M Bronstein. 2022. Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):657–668.

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*, pages 237–247.

Francesco Di Giovanni, Lorenzo Giusti, Federico Barbero, Giulia Luise, Pietro Lio, and Michael M Bronstein. 2023. On over-squashing in message passing neural networks: The impact of width, depth, and topology. In *International conference on machine learning*, pages 7865–7885. PMLR.

Herbert Edelsbrunner, John Harer, et al. 2008. Persistent homology-a survey. *Contemporary mathematics*, 453(26):257–282.

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. 2021. Twibot-20: A comprehensive twitter bot detection benchmark. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 4485–4494.

Chengqi Fu, Shuhao Shi, Yuxin Zhang, Yongmao Zhang, Jian Chen, Bin Yan, and Kai Qiao. 2023. Squeezegcn: Adaptive neighborhood aggregation with squeeze module for twitter bot detection based on gcn. *Electronics*, 13(1):56.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR.

Max Horn, Edward De Brouwer, Michael Moor, Yves Moreau, Bastian Rieck, and Karsten Borgwardt. 2021. Topological graph neural networks. *arXiv preprint arXiv:2102.07835*.

Haitao Huang, Hu Tian, Xiaolong Zheng, Xingwei Zhang, Daniel Dajun Zeng, and Fei-Yue Wang. 2024. Cgnn: A compatibility-aware graph neural network for social media bot detection. *IEEE Transactions on Computational Social Systems*.

Johanna Immonen, Amauri Souza, and Vikas Garg. 2023. Going beyond persistent homology using persistent homology. *Advances in neural information processing systems*, 36:63150–63173.

George Karypis and Vipin Kumar. 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1):359–392.

Feng Liu, Zhenyu Li, Chunfang Yang, Daofu Gong, Haoyu Lu, and Fenlin Liu. 2024. Segcn: a subgraph encoding based graph convolutional network model for social bot detection. *Scientific Reports*, 14(1):4122.

Lynnette Hui Xian Ng and Kathleen M Carley. 2025. A global comparison of social media bot and human characteristics. *Scientific Reports*, 15(1):10973.

T Konstantin Rusch, Michael M Bronstein, and Siddhartha Mishra. 2023. A survey on oversmoothing in graph neural networks. *arXiv preprint arXiv:2303.10993*.

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*.

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M Bronstein. 2021. Understanding over-squashing and bottlenecks on graphs via curvature. *arXiv preprint arXiv:2111.14522*.

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

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

Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The world wide web conference*, pages 2022–2032.

Xiaotang Wang, Yun Zhu, Haizhou Shi, Yongchao Liu, and Chuntao Hong. 2024. Graph triple attention network: A decoupled perspective. *arXiv preprint arXiv:2408.07654*.

A Waswani, N Shazeer, N Parmar, J Uszkoreit, L Jones, A Gomez, L Kaiser, and I Polosukhin. 2017. Attention is all you need. In *NIPS*.

Qitian Wu, Wentao Zhao, Zenan Li, David P Wipf, and Junchi Yan. 2022. Nodeformer: A scalable graph structure learning transformer for node classification. *Advances in Neural Information Processing Systems*, 35:27387–27401.

Qitian Wu, Wentao Zhao, Chenxiao Yang, Hengrui Zhang, Fan Nie, Haitian Jiang, Yatao Bian, and Junchi Yan. 2024. Simplifying and empowering transformers for large-graph representations. *Advances in Neural Information Processing Systems*, 36.

Yujie Xing, Xiao Wang, Yibo Li, Hai Huang, and Chuan Shi. 2024. Less is more: on the over-globalizing problem in graph transformers. *arXiv preprint arXiv:2405.01102*.

Chaoqi Yang, Ruijie Wang, Shuochao Yao, Shengzhong Liu, and Tarek Abdelzaher. 2020. Revisiting over-smoothing in deep gcns. *arXiv preprint arXiv:2003.13663*.

Ali Zia, Abdelwahed Khamis, James Nichols, Usman Bashir Tayab, Zeeshan Hayder, Vivien Rolland, Eric Stone, and Lars Petersson. 2024. Topological deep learning: a review of an emerging paradigm. *Artificial Intelligence Review*, 57(4):77.

## A Datasets

In this section, we analyze the basic structural properties of commonly used datasets in social bot detection, including the number of users (nodes), node degree distribution, number of edges, and graph density. Among these, graph density is a particularly informative metric. For directed graphs, it is defined as:

$$\text{Density} = \frac{E}{N \cdot (N-1)} \tag{24}$$

where $E$ denotes the number of edges and $N$ represents the total number of nodes. Graph density reflects the sparsity or denseness of connections in the graph: higher density indicates a more tightly connected and structurally complex graph, while lower density corresponds to a simpler, sparser topology.

### A.1 Twibot-20

The TwiBot-20 dataset selects a batch of node accounts as seed nodes placed at layer 0, considering users as nodes and following relationships as edges. The user information for layer $i+1$ is extracted from the users in layer $i$ along the edges, where up to 20 neighbor nodes are extracted for each follow and followed relationship. A total of three layers are extracted to form user clusters. This dataset construction method uses edges for expansion, forming a star-shaped structure centered around seed nodes, which tends to exhibit tree-like topological structures.

### A.2 MGTAB

The MGTAB dataset selects 100 seed nodes and constructs the user network using breadth-first search. At the same time, each user acquires up to 10,000 tweets, from which mention, reply, quote, and other relations are extracted to construct a multi-relational network of users. Additionally, MGTAB undergoes cleaning by discarding users irrelevant to the target topic and outlier accounts without following relationships, thereby constructing a compact topological dataset strongly related to the theme.

### A.3 TwiBot-22(subgraph)

TwiBot-22 is the largest dataset in the field of social bot detection to date, encompassing four types of entities—user, tweet, list, and hashtag—as well as a total of 14 relationship types, including follower, following, and mention. However, when performing bot account detection, only the user entity and its internal follower and following relationships are directly utilized for analysis. We observe that this approach results in numerous outliers, specifically accounts without any followers or friends. This limitation leads to the emergence of a significant number of isolated nodes in the graph structure, which are essentially accounts lacking both followers and friends. To enhance the effectiveness and accuracy of data analysis, we adopt a processing method inspired by MGTAB, where such unassociated users are removed to refine the dataset. Consequently, we construct a well-connected subgraph collection, referred to as TwiBot-22(subgraph).

## B Baselines

We compared our model with some state-of-the-art models on graph neural networks to verify the effectiveness of our optimizations and improvements for graph neural networks.

- **GCN**: Utilizes graph convolutional neural networks to aggregate messages within the neighborhood of nodes to learn node features.

- **GAT**: Employs Transformer mechanisms to autonomously learn edge weights for bot detection.

- **RGCN**: Constructs heterogeneous graphs to represent network structures and utilizes relational graph convolutional neural networks to learn node features for bot detection.

- **RGT**: Uses relational graph attention networks and captures bot node features under different relationships through semantic attention mechanisms for social bot detection.

- **Nodeformer**: A Graph Transformer based on global attention that implements efficient pairwise node signal transmission using a kernelized Gumbel-Softmax operator, addressing shortcomings of traditional graph neural networks in handling overfitting, heterogeneity, long-range dependencies, and missing graph structures.

## C Layer Placement Study

As indicated in (Borgwardt et al., 2020), node features often contain a substantial amount of discriminative information, which may to some extent suppress the additional gains provided by the graph

Table 5: Position Analysis of TOGL: Performance comparison under different insertion positions across datasets.

| TOGL Position | MGTAB | | TwiBot-20 | | TwiBot-22(subgraph) | |
|---|---|---|---|---|---|---|
| | ACC (%) | F1 (%) | ACC (%) | F1 (%) | ACC (%) | F1 (%) |
| Without TOGL | 86.80 ± 0.48 | 76.02 ± 1.77 | 85.53 ± 1.14 | 86.81 ± 1.48 | 90.05 ± 0.08 | 52.27 ± 2.54 |
| Before GNN | 85.51 ± 0.70 | 75.15 ± 1.39 | 86.15 ± 0.14 | 87.66 ± 0.16 | 90.02 ± 0.23 | 51.31 ± 2.72 |
| After GNN | **86.90 ± 0.69** | **76.86 ± 1.40** | **86.61 ± 0.17** | **88.12 ± 0.23** | **90.39 ± 0.09** | **51.83 ± 1.83** |

Table 6: Graph Structural Statistics of Bot Detection Datasets

| Feature | Cresci-15 | TwiBot-20 | MGTAB | TwiBot-22(subgraph) |
|---|---|---|---|---|
| Real Users | 1,950 | 5,237 | 7,451 | 612,328 |
| Bot Accounts | 3,351 | 6,589 | 2,748 | 81,431 |
| Total Nodes | 5,301 | 229,580 | 10,199 | 693,759 |
| Maximum Degree | 416 | 20 | 8,383 | 2,229 |
| Average Degree | 2.68 | 0.99 | 166.69 | 10.79 |
| Number of Edges | 14,220 | 227,979 | 1,700,108 | 3,743,633 |
| Graph Density | 0.001012 | 0.000009 | 0.032691 | 0.000008 |

structure. To explore how the position of the topological layer affects model performance, we place it either before or after the message-passing GNN layers. Experimental results, as shown in Table 5, reveal a consistent trend across all three datasets: placing the topological layer before the GNN layers leads to a decline in performance, whereas positioning it after results in significantly improved model effectiveness.

## D Parameter Study

**Numbers of clusters:** Due to the high computational cost associated with increasing the number of clusters, we limit the number of clusters to within 500 and divide them at intervals of 100 to investigate the potential relationship between the number of clusters and the dataset structure. As shown in Figure 4, larger datasets are more prone to over-globalization issues, necessitating more clusters to focus attention on smaller regions. Furthermore, datasets with more complex structures (e.g., MGTAB) require more clusters to decompose local information effectively.

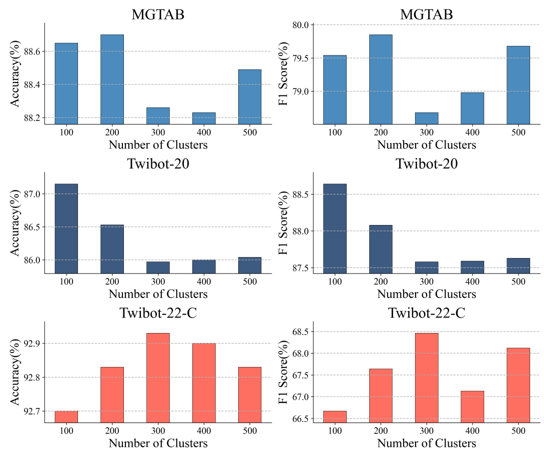


Figure 4: ACC and F1-score on the datasets for varying numbers of clusters