

# CACL: Community-Aware Heterogeneous Graph Contrastive Learning for Social Media Bot Detection

Sirry Chen<sup>♣\*</sup>, Shuo Feng<sup>♣\*</sup>, Songsong Liang<sup>♣</sup>, Chen-Chen Zong<sup>♣</sup>, Jing Li<sup>♡</sup>, Piji Li<sup>♣†</sup>

<sup>♣</sup>Nanjing University of Aeronautics and Astronautics

<sup>♡</sup>The Hong Kong Polytechnic University

{sirrychen, pjli}@nuaa.edu.cn

## Abstract

Social media bot detection is increasingly crucial with the rise of social media platforms. Existing methods predominantly construct social networks as graph and utilize graph neural networks (GNNs) for bot detection. However, most of these methods focus on how to improve the performance of GNNs while neglecting the community structure within social networks. Moreover, GNNs based methods still face problems such as poor model generalization due to the relatively small scale of the dataset and over-smoothness caused by information propagation mechanism. To address these problems, we propose a **Community-Aware Heterogeneous Graph Contrastive Learning** framework (CACL), which constructs social network as heterogeneous graph with multiple node types and edge types, and then utilizes community-aware module to dynamically mine both hard positive samples and hard negative samples for supervised graph contrastive learning with adaptive graph enhancement algorithms. Extensive experiments demonstrate that our framework addresses the previously mentioned challenges and outperforms competitive baselines on three social media bot benchmarks.

## 1 Introduction

Social media bots, widely found within social networks, are software systems designed to interact with humans (Orabi et al., 2020), e.g., 9% to 15% active users on Twitter exhibit bot-like behaviors (Varol et al., 2017). Although social media bots have brought convenience to people’s daily life, including news broadcasts, online customer service, etc., their negative applications, such as online disinformation (Cresci et al., 2023), election interference (Cresci, 2020; Ferrara, 2017), extremism advocacy (Berger and Morgan, 2015), seem to be more widespread and absolutely cannot be

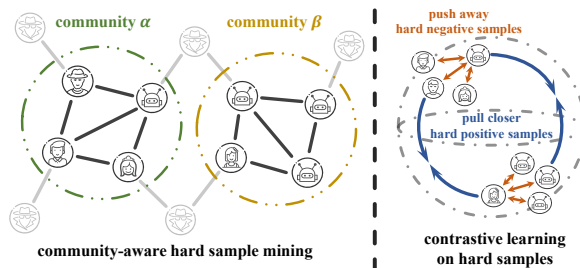


Figure 1: Illustration of considering diff-class nodes within one community as hard negative samples while same-class nodes between communities as hard positive samples. Based on these, graph contrastive learning is employed to push hard negative samples away and pull hard positive samples closer on the hypersphere.

overlooked. To solve the above issues, numerous methods have been proposed to detect social media bots efficiently.

Existing works for social media bot detection primarily consist of feature based and graph based methods. In recent years, because feature engineering (Yang et al., 2020; Feng et al., 2021a; Dehghan et al., 2023) has been proven to be incapable of addressing the challenge of generalization (Feng et al., 2022b), graph convolutional neural networks (GCNs) (Kipf and Welling, 2016) based models have been designed to leverage graph structure information. While homogeneous graphs (Alhosseini et al., 2019) fail to model diverse relationships between social accounts, such as following and commenting, methods based on heterogeneous graphs, e.g., heterogeneous GCNs (Zhao et al., 2020), relational GCNs (Feng et al., 2021c), relational graph transformer (Feng et al., 2022a), etc., can better extract node features by capturing information from various types of relationships.

However, most of these efforts primarily focus on improving the performance of node feature extraction by designing sophisticated GNNs, while neglecting the inherent characteristic of social networks, i.e., social networks naturally divide into

\* Equal contribution.

† Corresponding author.

groups of users with denser connections inside each group and fewer connections between different groups (Nguyen et al., 2014; Cherifi et al., 2019; Liu et al., 2023). For example, under a politically related topic, users tend to interact more frequently with each other than with others who are not interested in this topic (new, 2004). Intuitively, these users form a political community. And in this community, to influence public opinion or achieve other objectives, bots often share political opinions and interact with humans. As a result, they exhibit shared interests and characteristics with humans in the same community, while bots from other communities are relatively different from the humans in this community (Tu et al., 2019). This phenomenon makes bots from other communities can be easily distinguished from users in this political community, once we identify the community structures.

While leveraging these community structures can facilitate bot detection, it also introduces problems. When we treat users as nodes, due to the over-smoothing problem (Chen et al., 2020), GNNs based methods will result in highly similar representations for nodes in the same community, and different representations for nodes in different communities. Therefore, diff-class nodes in one community tend to have short distances in the feature space, while same-class nodes in different communities tend to have long distances between them (Zhou et al., 2020). As illustrated in Figure 1, we define diff-class nodes in one community as hard negative samples and same-class nodes in different communities as hard positive samples. These hard samples bring difficulties to bot detection.

To address the above challenges, we propose the **Community-Aware Heterogeneous Graph Contrastive Learning** framework (i.e., **CACL**)<sup>1</sup>, a graph contrastive learning framework capable of perceiving community structures. In general, this framework is divided into graph level and node level components. Firstly, we design a community-aware module to perform community detection and mine both hard negative and positive samples. Then, supervised graph contrastive learning is introduced to handle these hard samples, by guiding GNNs model to focus on subtle differences among hard negative samples and push them away, and reasonably disregard the distinct features among hard positive samples and pull them closer. Besides, the underlying feature extraction networks

in community-aware module are shared with the networks in contrastive learning module, allowing us to perform dynamic and adaptive hard sample mining. In this manner, by continuously updating model parameters and mining new hard samples, the representations of same-class nodes are consistently gathered on the hypersphere while those of diff-class are separated, which is beneficial to classification. To construct contrastive graphs, we introduce link prediction, synonymy substitution, and node feature shifting that are adaptive to graph topology structure, text information, and users' attributes, respectively. We summarize our main contributions as follows:

- We design a community-aware module to dynamically extract community structures within social networks, and perform both hard negative and positive sample mining based on these structures.
- We propose a graph contrastive learning framework with community structure-aware capacity, and jointly perform data augmentation on topology, text and attribute levels to generate contrastive graphs.
- Extensive experiments on three social media bot detection datasets demonstrate our framework can significantly improve the performance of several GNNs backbones, which bears out the effectiveness and superiority of **CACL** framework.

## 2 Related work

### 2.1 Graph-based Bot Detection

Traditional neural networks (Kudugunta and Ferrara, 2018; Wu et al., 2021) struggle to directly handle non-Euclidean structured data while GNNs provide an effective way. Alhosseini et al. (2019) conceptualize users as nodes and construct a graph with single type of relationships between users. Feng et al. (2022a) utilize the Relation Graph Transformer which employs a semantic attention mechanism to aggregate information from different types of edges. Yang et al. (2023) construct a heterogeneous graph with diverse relationships between these users and their texts. Despite their success, existing methods still face problems such as poor model generalization and over-smoothness (Ma et al., 2021). To mitigate the above issues, our approach leverages the community-aware module to mine both hard negative and positive samples and employs graph contrastive learning to deal with these samples.

<sup>1</sup><https://github.com/SirryChen/CACL>

## 2.2 Community Detection on Social Networks

Early methods, such as Louvain algorithm (Blondel et al., 2008), utilize only graph structure information and aim to maximize modularity during community detection. In recent years, deep neural networks have been introduced to combine node features with graph structure information. Methods such as Markov Random Fields (He et al., 2020), Graph Autoencoders (GAE) (Wang et al., 2020), and Variational Graph Autoencoders (VGAE) (Salha-Galvan et al., 2022) are employed to capture graph structure information and conduct unsupervised community detection. Different from Tan et al. (2023), who split whole social networks into several vast communities, we follow the original definition of community and consider each community as a subgraph.

## 2.3 Graph Contrastive Learning

Contrastive learning originates in the field of image processing and has been widely applied to non-Euclidean graph processing in recent years. GRACE (Zhu et al., 2020) employs feature mask and edge removal to generate augmented views and regards the same node in two views as a positive pair. CBD (Zhou et al., 2023a) adopts contrastive learning in the pre-training stage to extract knowledge from unlabeled data and uses a consistency loss in the fine-tuning stage to improve the detection performance. BotSCL (Wu et al., 2023) treats nodes with the same labels as positive samples and other nodes as negative samples. However, these strong setting of hard samples may lead to issues such as model instability and poor generalization.

# 3 Methodology

## 3.1 Overview

**Problem Definition.** We construct a heterogeneous graph  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R}\}$  where users and texts (e.g., tweets) are treated as nodes  $\mathcal{V} = \{v_i \mid i = 1, 2, \dots, n\}$  with various relationships serving as edges  $\mathcal{E} = \{e_{ij} \mid v_i, v_j \in \mathcal{V}\}$ .  $\mathcal{A} = \{a_i \mid v_i \in \mathcal{V}\}$  collects the types of nodes, and  $\mathcal{R} = \{r_{ij} \mid e_{ij} \in \mathcal{E}\}$  collects the types of edges. Social media bot detection task is to learn the distribution of nodes to identify bots among users and apply it to real-world networks.

Figure 2 displays an overview of our proposed framework **CACL**. In general, we divide it into coarse-grained and fine-grained operations, corresponding to graph level and node level, respec-

tively. Firstly we perform community detection on the user graph to identify community structures and treat each community as a subgraph. We then perform three graph augmentation methods on subgraphs. Subsequently, we find two matched subgraphs and conduct graph contrastive learning on them to obtain the optimal user representations and finally classify users into bots and humans.

## 3.2 Community-Aware Module

For each input graph  $\mathcal{G}$ , we first extract user graph  $\mathcal{G}_u = \{\mathcal{V}_u, \mathcal{E}_u, \mathcal{A}_u, \mathcal{R}_u\}$  to simplify the graph structure and generate the adjacency matrix  $A$  which disregards different edge types between users. The initial features of users encompass metadata, categorical metadata, descriptions, and posted tweets. We apply z-score normalization to numerical features and use RoBERTa (Liu et al., 2019) to encode textual features. We denote user  $v_i$ 's feature vector as  $\mathbf{r}_i$  and preprocess it with a fully connected layer:

$$\mathbf{x}_i = \sigma(\mathbf{W}_p \mathbf{r}_i), \quad (1)$$

where  $\mathbf{W}_p$  is learnable parameter, and  $\sigma$  denotes Leaky-ReLU without further notice.

To incorporate graph structural information into node features and aggregate neighbors' information, we use multi-layer GCNs as encoder to encode the user graph as follows:

$$\mathbf{h}^{l+1} = \sigma(\mathbf{A} \mathbf{h}^l \mathbf{W}_h^l), \quad (2)$$

where  $\mathbf{h}_i^0 = \mathbf{x}_i$  is the input of the first layer,  $l$  denotes the  $l$ -th GCNs layer, and  $\mathbf{W}_h^l$  is learnable parameter.  $\mathbf{h}_i^L$  is the final representation of  $v_i$ .

After obtaining users' representations encoded by  $L$ -layer GCNs, we assign the cosine similarities  $\cos(\mathbf{h}_i^L, \mathbf{h}_j^L)$  as weights to corresponding edge  $e_{ij}$ . Then we traverse edges in descending order of weights, consecutively cluster the nodes connected by each edge into same community, and merge overlapped communities until the number of communities equals  $k$ . To follow the traditional definition of community, i.e., the internal degree of each vertex is greater than its external degree (Fortunato, 2010),  $k$  is given by Louvain algorithm (Blondel et al., 2008) which relies entirely on the inherent graph structure without node features. In this manner, the user graph is partitioned into subgraphs and each subgraph represents a community. Users in the same subgraph have similar representations, and users in different subgraph have different representations, which satisfies the previously definition of hard samples.

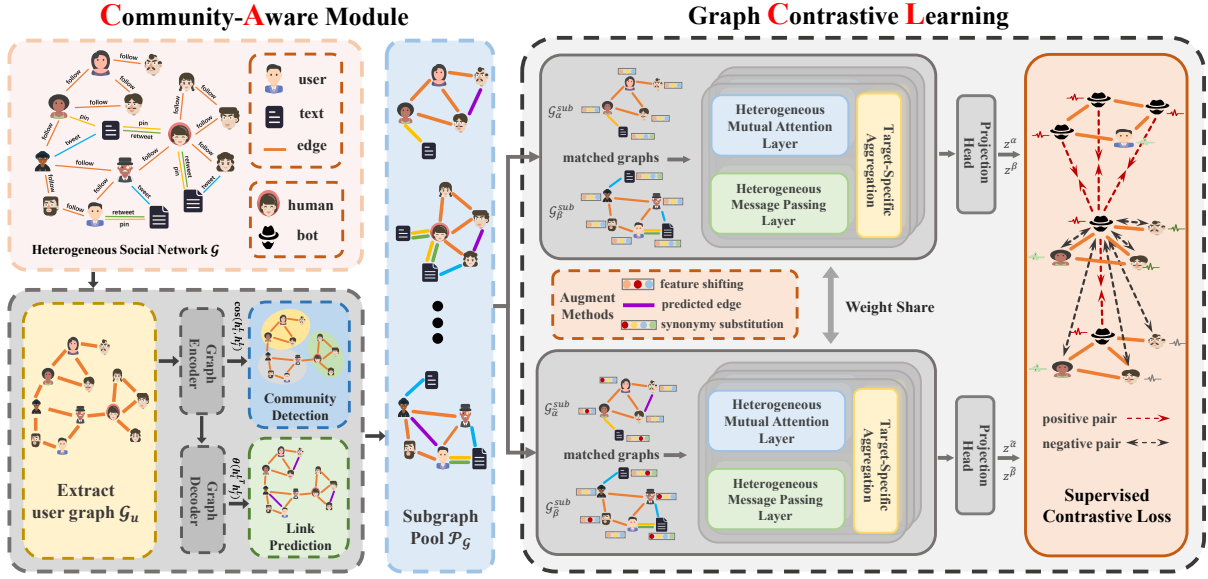


Figure 2: **Overview of our proposed framework CACL.** In graph level, we employ community-aware module to split social networks into communities and mine hard negative and positive samples, creating a subgraph pool. From this pool, we constantly select two matched subgraphs. Then, in node level, we utilize three graph augmentation methods to generate augmented graph and introduce graph contrastive learning to handle hard samples.

To ensure that the community-aware module can simultaneously accomplish community detection task, as well as link prediction used in Section 3.3, we refer to (Salha-Galvan et al., 2022) and employ a modularity-inspired loss function  $\mathcal{L}_{CA}$  to pre-train community-aware module:

$$\mathcal{L}_{CA} = \mathcal{L}_G + \mathcal{L}_M, \quad (3)$$

$$\mathcal{L}_G = -\frac{1}{n^2} \sum_{e_{ij} \in \mathcal{E}_u} \mathbf{CE}(A_{ij}, \hat{A}_{ij}), \quad (4)$$

$$\mathcal{L}_M = -\frac{\beta}{2m} \sum_{i,j=1}^n \left[ A_{ij} - \frac{d_i d_j}{2m} \right] e^{-\gamma \| \mathbf{h}_i^L - \mathbf{h}_j^L \|^2}, \quad (5)$$

where  $\mathbf{CE}$  denotes cross-entropy function and  $d_i$  is the degree of node  $v_i$ .  $\mathcal{L}_G$  is a standard reconstruction loss of graph auto-encoder, while  $\mathcal{L}_M$  is a community-aware loss inspired by modularity. In Equation 5,  $n$  denotes the number of users,  $m$  denotes the number of edges, and the exponential term, which takes values in the range  $[0, 1]$ , serves as a softened counterpart of community indicator  $\delta_{ij} \in \{0, 1\}$  in original modularity function, where  $\delta_{ij}$  equals 1 when  $i = j$  and 0 otherwise. The hyperparameter  $\beta$  is used to balance the two loss terms and  $\gamma$  regulates the magnitude of the exponential term.

In essence,  $\mathcal{L}_M$  protects graph structure by ensuring structurally adjacent nodes are more likely

to be assigned to one community, while  $\mathcal{L}_G$  aims to preserve the performances on link prediction.

### 3.3 Graph Augmentation

Through community-aware module, we split user graph into a series of user subgraphs. To maintain graph structure, we restore the user subgraphs to original heterogeneous graph and form a subgraph pool  $\mathcal{P}_G = \{\mathcal{G}_i^{sub}\}$ . In order to improve the effectiveness of graph contrastive learning, we generate augmented graph  $\mathcal{G}_\alpha^{sub}$  after processing original graph  $\mathcal{G}_\alpha^{sub}$  with three methods, including node feature shifting, link prediction, and synonymy substitution.

**Node Feature Shifting.** To introduce perturbations on less important node features while preserving crucial ones, we adaptively mask certain dimensions by calculating feature importance using PageRank algorithm (Page et al., 1999; Zhu et al., 2021). Formally, we sample a mask vector  $\mathbf{m}_a \in \mathbb{R}^{d_a}$  for nodes of node type  $a$ , where each dimension  $\mathbf{m}_o$  is independently drawn from Bernoulli distribution  $Bern(1 - p_o^f)$ ,  $o = 1, 2, \dots, d_a$ , and  $d_a$  is corresponding node feature dimension. We assume that a higher feature value, such as high following number, indicates a greater significance of attribute, which holds true for both discrete and continuous node attributes. So we calculate the



weight of dimension  $o$  as:

$$w_o^f = \log \sum_{v_i \in \mathcal{V}} |\mathbf{x}_i| \cdot \rho_c(v_i), \quad (6)$$

where  $\rho_c(v_i)$  denotes PageRank centrality of node  $v_i$ . Then we can obtain the probability representing feature importance:

$$p_o^f = \min \left( \frac{w_{max}^f - w_o^f}{w_{max}^f - w^f} \cdot p_f, p_\tau \right), \quad (7)$$

where hyperparameter  $p_f$  controls the magnitude of probability while  $p_\tau$  is a truncation value. Finally, we get  $v_i$ 's augmented feature vector  $\tilde{\mathbf{r}}_i = \mathbf{r}_i \circ \mathbf{m}_{a_i}$ , where  $\circ$  is Hadamard product.

**Link Prediction.** To balance in-degree and out-degree of nodes (Zhou et al., 2023b) and enhance the model's robustness against interference from changes in relationships between nodes, we perform link prediction and connect potentially existing edges. For consistency with previous works, we use inner product to act as decoder in the community-aware module and utilize the output as the probability of an edge existing between user  $v_i$  and  $v_j$ ,

$$\mathcal{E}_{\text{pred}} = \{e_{ij} \mid \theta(\mathbf{h}_i^{L,T} \mathbf{h}_j^L) > p_e\}, \quad (8)$$

where  $\theta$  is the sigmoid function,  $p_e$  is hyperparameter used to control the number of edges to prevent irreversible disruption of graph structure caused by excessive edges. Then the augmented edge set  $\mathcal{E}_{\tilde{\alpha}} = \{\mathcal{E}_\alpha, \mathcal{E}_{\text{pred}}\}$ .

**Synonymy Substitution.** Social networks contain a considerable amount of textual information. Through using WordNet (Miller, 1995) pre-trained with twitter corpus, we replace words with their synonyms to enhance the texts without altering its fundamental semantics.

### 3.4 Graph Convolutional Networks

Different from homogenous networks, heterogeneous social networks consist of nodes and edges of different types, which requires GNNs to be capable of diverse embedding dimensions and different edge types in information propagation. GNNs methods such as SAGE (Hamilton et al., 2017), GAT (Velickovic et al., 2017), HGT (Hu et al., 2020) can be employed by our CACL framework.

We first preprocess user  $v_i$ 's initial feature  $\mathbf{r}_i$  with a fully connected layer:

$$\hat{\mathbf{x}}_i = \sigma(\mathbf{W}_q^{a_i} \mathbf{r}_i), \quad (9)$$

where  $\mathbf{W}_q^{a_i}$  is learnable parameter for node type  $a_i$ .  $\mathbf{W}_q^u$  for user nodes is shared with  $\mathbf{W}_p$  in community-aware module to perform dynamic hard sample mining. We feed  $\hat{\mathbf{x}}_i$  into GNNs model and denote that the  $L$ -th layer GNNs model outputs the target node hidden state  $\mathbf{H}_i^L$  which contains information from all its neighbors of different types as well as its own feature in the view of community  $\mathcal{G}_\alpha^{sub}$ . Therefore, by utilizing these node representations, we can proceed with contrastive learning.

### 3.5 Community-Aware Contrastive Loss

Referring to traditional graph contrastive learning frameworks (You et al., 2020; Li et al., 2022), we employ a two-layer perceptron as a projection head to obtain representations for each user  $v_i$ :

$$\mathbf{z}_i = \mathbf{W}_2 (\mathbf{W}_1 \mathbf{H}_i^L), \quad (10)$$

where  $\mathbf{W}_1$  and  $\mathbf{W}_2$  are learnable parameters. For original subgraph  $\mathcal{G}_\alpha^{sub}$  and augmented subgraph  $\mathcal{G}_{\tilde{\alpha}}^{sub}$ , we can get two representations  $\mathbf{z}_i^\alpha$  and  $\mathbf{z}_i^{\tilde{\alpha}}$  for the same user node  $v_i$ .

To solve the false negatives problems caused by unsupervised contrastive learning (Khosla et al., 2020), Wu et al. (2023) leverage annotated information during graph contrastive learning. They try to pull closer all of same-class nodes and push away all of diff-class nodes in constructed graph. In realistic scenarios, however, the differences between different users are often substantial even in the same community. If we unreasonably force high-dimensional semantic features of all different same-class nodes to converge to same point on the hypersphere, it may lead to issues such as model instability and poor generalization. Therefore, we modify the traditional loss function and choose to only push away hard negative samples, which are diff-class nodes in the same community, while pull closer hard positive samples, which are same-class nodes in the different communities just as showing in Figure 1.

Given one subgraph  $\mathcal{G}_\alpha^{sub}$ , we firstly find its matched subgraph. By computing an average embedding vector for every subgraph and then cosine similarity between each pair of subgraphs in the subgraph pool  $\mathcal{P}_G$ . Its matched subgraph  $\mathcal{G}_\beta^{sub}$  has the lowest similarity with  $\mathcal{G}_\alpha^{sub}$  and thus the positive samples in  $\mathcal{G}_\beta^{sub}$  is hard enough intuitively. We then consider target node representation  $\mathbf{z}_i^\alpha$  in  $\mathcal{G}_\alpha^{sub}$ ,

the modified contrastive loss function is as follows:

$$\begin{aligned} \mathcal{L}_i^\alpha &= -\frac{1}{N_\alpha} \sum_{v_i \in \mathcal{V}_\alpha} \log \frac{s_i^{self} + s_i^{pos}}{s_i^{self} + s_i^{pos} + s_i^{neg}}, \quad (11) \\ s_i^{self} &= e^{\cos(\mathbf{z}_i^\alpha, \mathbf{z}_i^{\tilde{\alpha}})}, \\ s_i^{pos} &= \frac{1}{2} \sum_{v_j \in \{\mathcal{V}_\beta, \mathcal{V}_{\tilde{\beta}}\}} \delta_{ij} e^{\cos(\mathbf{z}_i^\alpha, \mathbf{z}_j^{\beta, \tilde{\beta}})}, \\ s_i^{neg} &= \frac{1}{2} \sum_{v_j \in \{\mathcal{V}_\alpha, \mathcal{V}_{\tilde{\alpha}}\}} \hat{\delta}_{ij} e^{\cos(\mathbf{z}_i^\alpha, \mathbf{z}_j^{\alpha, \tilde{\alpha}})}, \end{aligned}$$

where  $\hat{\delta}_{ij}$  is opposite of  $\delta_{ij}$ .  $s_i^{self}$  represents the similarities of target node  $v_i$  in different views,  $s_i^{pos}$  and  $s_i^{neg}$  denote the similarities between target node and its hard positive samples and hard negative samples, respectively. After performing the same operations on the nodes in  $\mathcal{G}_\alpha^{sub}$ , we obtain the final contrastive loss:

$$\mathcal{L}_{contrast} = \sum_{\mathcal{G}_\alpha^{sub} \in \mathcal{P}_\mathcal{G}} \frac{1}{2N_\alpha} \sum_{i=1}^{N_\alpha} \mathcal{L}_i^\alpha + \mathcal{L}_i^{\tilde{\alpha}}, \quad (12)$$

where  $N_\alpha$  is the number of users in  $\mathcal{G}_\alpha^{sub}$ . We incorporate this contrastive loss function as a regularization term into the overall model loss function:

$$\mathcal{L}_{CL} = \lambda \cdot \mathcal{L}_{contrast} + (1 - \lambda) \cdot \mathcal{L}_{classify}, \quad (13)$$

where the second term represents the cross-entropy classification loss, and  $\lambda$  is weight adjustment hyperparameter.

### 3.6 Training Strategies

In this paper, to mitigate the impact of low-quality community partitions on graph contrastive learning during cold start, we initially perform unsupervised pre-training on community-aware module, using Equation 3. In the training process, we iteratively select a certain number of adjacent users to form an input graph. After obtaining the subgraph pool, we continuously retrieve two matched subgraphs from the subgraph pool, and then employ graph contrastive learning for end-to-end model training on these two matched subgraphs using Equation 13.

## 4 Experimental Setups

**Datasets.** Our method is graph-based and heterogeneous-aware, which requires information of diverse relation types between nodes in datasets. **Cresci-15** (Cresci et al., 2015), **Twibot-20** (Feng et al., 2021b) and **Twibot-22** (Feng et al., 2022b)

all contain various relationships among users as well as between users and their tweets which are essential to build heterogeneous networks. We follow the same splits provided in the benchmarks to mitigate the potential impact on experiment results.

**Baselines.** To demonstrate the effectiveness of our proposed framework **CACL**, we choose three backbone models, i.e., GCN (Kipf and Welling, 2016), GAT (Velickovic et al., 2017), HGT (Hu et al., 2020), as convolution layers and compare them with previous representative works:

- **SimpleHGN** (Lv et al., 2021) designs residual connections and output embeddings normalization, which is suitable for heterogeneous graphs.
- **BotRGCN** (Feng et al., 2021c) leverages relational graph convolutional networks, enhancing its ability to capture diverse disguises of robots.
- **RGT** (Feng et al., 2022a) utilizes a multi-attention mechanism and a semantic attention network to learn and aggregate user representations under each relationship, respectively.
- **DeeProBot** (Hayawi et al., 2022) is designed with LSTM units to process the mixed input, extracting temporal information from user text.
- **EvolveBot** (Yang et al., 2013) extracts features such as betweenness centrality and clustering coefficient from graph structure.
- **BGSRD** (Guo et al., 2022) utilizes Roberta to process user descriptions and make prediction with the combination of BERT and GAT outputs.
- **SATAR** (Feng et al., 2021a) considers the follower number as self-supervised label to pre-train models and then fine-tune parameters.
- **RoBERTa** (Liu et al., 2019) is a pre-trained language model based on the Transformer architecture, employed by Twibot-22 benchmark.

**Implementation.** We set  $\lambda$  to 0.9,  $\tau$  to 0.07, learning rate to 0.0001, layer number of all GNNs models to 2, and dropout rate to 0.5. In each epoch, we iteratively select 2000 adjacent users for Twibot-22 dataset and 1000 for other datasets. We conduct all experiments on 2 NVIDIA RTX A5000 GPUs.

## 5 Results and Analysis

### 5.1 Main Results

We evaluate **CACL** framework with three backbones and other representative baselines on the three social media bot detection benchmarks and

Method	Cresci-15			Twibot-20			Twibot-22		
	Accuracy	F1-score	MCC	Accuracy	F1-score	MCC	Accuracy	F1-score	MCC
BotRGCN	94.71	95.85	89.06	<u>84.43</u>	85.79	67.17	73.90	<u>48.30</u>	31.01
DeeProBot	72.35	81.61	44.26	77.34	80.32	54.40	74.11	12.73	14.12
RGT	96.27	<u>97.04</u>	<u>92.25</u>	84.02	86.07	68.08	74.49	45.36	29.64
EvolveBot	92.18	90.07	72.77	65.83	69.75	30.79	71.09	14.09	13.38
SimpleHGN	93.13	94.68	85.91	83.93	86.33	68.40	72.53	<b>49.59</b>	30.74
RoBERTa	<u>97.01</u>	95.86	89.55	75.55	73.09	42.15	72.07	20.53	19.35
BGSRD	87.78	90.80	74.92	66.36	70.05	32.28	71.88	21.14	20.32
SATAR	92.71	94.55	85.61	84.02	85.74	67.83	/	/	/
GAT	93.52	94.99	86.78	82.24	84.93	64.95	74.17	43.75	28.12
+CACL	94.51	95.42	88.71	83.52	86.39	68.56	74.50	44.27	28.94
SAGE	93.92	95.26	87.48	83.00	85.48	66.38	74.39	46.47	30.25
+CACL	<b>97.65</b>	<b>98.12</b>	<b>95.10</b>	83.60	<u>86.53</u>	<u>68.95</u>	<b>75.38</b>	47.45	<u>32.27</u>
HGT	92.24	94.27	84.83	84.02	86.35	68.48	<u>75.07</u>	43.13	29.18
+CACL	95.88	96.74	91.45	<b>85.12</b>	<b>87.28</b>	<b>70.75</b>	<u>75.07</u>	48.27	<b>32.39</b>

Table 1: **Main results of CACL compared with various baselines on three datasets.** **Bold** and underline indicate the highest and second highest performance, respectively, while "/" indicates that the baseline is not scalable to the corresponding dataset. GAT, SAGE and HGT are three backbones we use as the graph convolutional model in our framework.

present the results in Table 1. The results demonstrate that:

- Although previous methods have made great efforts, the improvements of models' performances are not significant compared with traditional GNNs methods.
- Our proposed CACL framework markedly enhance the performance of various convolutional networks, including GAT, SAGE and HGT, and can even outperform the state-of-the-art.
- CACL framework with different backbones presents significant improvement compared with the original methods, which elucidates that contrastive learning and the community-aware module help GNNs models to better capture node information.

## 5.2 Contrastive Learning Study

To compare diverse contrastive learning loss functions, we experiment with three loss functions while keeping other parameters constant. **Unsupervised contrastive loss** (You et al., 2020) tries to push the representations of all nodes away from each other and pull representations of every node in different views together without using the label information. **Supervised contrastive loss** (Khosla et al., 2020; Wu et al., 2023) aims to push the representations of diff-class nodes away from each other

CL Loss	Accuracy	F1-score	MCC
unsuper	84.27	86.75	69.40
super	84.78 <sub>↑0.51</sub>	86.95 <sub>↑0.20</sub>	69.97 <sub>↑0.57</sub>
static	84.45 <sub>↑0.18</sub>	86.74 <sub>↓0.01</sub>	69.42 <sub>↑0.02</sub>
dynamic	85.12 <sub>↑0.85</sub>	87.28 <sub>↑0.53</sub>	70.75 <sub>↑1.35</sub>

Table 2: **Performances of HGT model with different contrastive loss function**, including unsupervised and supervised contrastive loss function, whose definitions of hard samples are different from ours, and our modified loss function in CACL framework with static or dynamic community-aware module.

while pull representations of same-class nodes together, which are conducted across original graph and augmented graph. Our proposed CACL framework utilizes the inherent community structure to mine hard positive and negative samples for bot detection.

The comparison results in Table 2 show that, compared with unsupervised contrastive loss function, other loss functions can significantly improve the performance of HGT model. Our proposed community-aware contrastive loss function outperforms the traditional supervised contrastive loss function, which indicates that mining hard samples under the consideration of community structure is effective for contrastive learning. However, when we freeze community-aware module and thus it

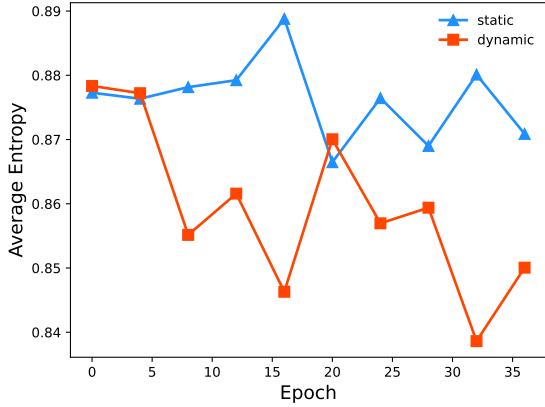


Figure 3: **Contrast of community entropy change trends** between static and dynamic community-aware module. Static means freezing community-aware module and thus cannot dynamically mine hard samples during training process. Average entropy is a metric for measuring the imbalance of categories.

remains static during the training process, the improvement decreases significantly, showing the importance of dynamism of this module.

### 5.3 Community-aware Study

To investigate the internal mechanism of how community-aware module helps model to focus on hard samples and classify them correctly, we use information entropy, which is adopted in decision trees (Quinlan, 1986), to measure the level of category disorder in each community we detect in every batch. To demonstrate the overall trend, we take average of entropy in each epoch and make contrast between dynamic and static community-aware module. The results shown in Figure 3 demonstrate that dynamic community-aware module tend to cluster nodes of same labels to the same community, which proves it can dynamically capture the community structures within social networks as parameters are continuously updated in the training process, while the static module fails.

To further qualitatively analyze whether the community detection result is meaningful, and how the contrastive learning design helps to deal with the hard samples, we visualize the changes of average cosine similarity among nodes in one community and nodes in different communities. The results shown in Figure 4 prove that the similarities between positive samples continuously rise and those between negative samples decrease, which means same-class nodes are pulled together and diff-class nodes are pushed apart. The results also

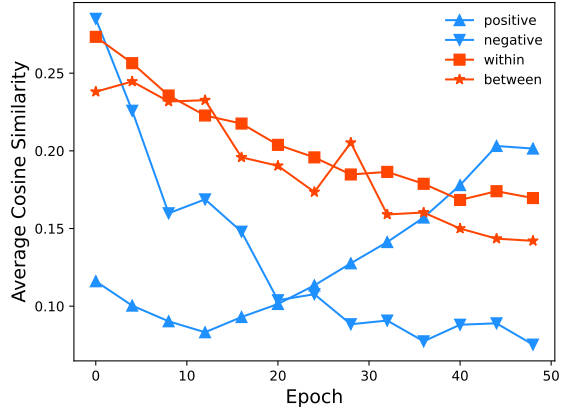


Figure 4: **Cosine similarity change trends**. Positive and negative represent cosine similarities between positive samples and between negative samples, respectively. Within and between represent cosine similarities within one community and between different communities, respectively.

Ablation	Accuracy	F1-score	MCC
<b>CACL</b>	85.12	87.28	70.75
w/o SS	84.78 $\downarrow$ 0.34	87.19 $\downarrow$ 0.09	87.19 $\downarrow$ 0.24
w/o LP	84.45 $\downarrow$ 0.67	86.98 $\downarrow$ 0.30	70.01 $\downarrow$ 0.74
w/o FS	85.12 $\downarrow$ 0.00	87.09 $\downarrow$ 0.19	70.42 $\downarrow$ 0.33
w/o TX	83.34 $\downarrow$ 1.78	86.51 $\downarrow$ 0.77	69.14 $\downarrow$ 1.61
w/o MT	82.92 $\downarrow$ 2.20	86.10 $\downarrow$ 1.18	67.97 $\downarrow$ 2.78
w/o HG	82.75 $\downarrow$ 2.37	85.28 $\downarrow$ 2.00	65.87 $\downarrow$ 4.88
w/o CA	84.36 $\downarrow$ 0.76	86.44 $\downarrow$ 0.84	68.87 $\downarrow$ 1.88

Table 3: **Ablation study of CACL**. Experiments mainly focus on there aspects. SS, LP, FS symbolizes three data augmentation methods, which are synonymy substitution, link prediction and feature shifting, respectively. TX, MT, HG indicates three different kinds of information in social networks, which are users’ texts, metadata and heterogeneity of graph. CA denotes the community-aware hard samples mining.

show that similarities of nodes within and between communities decrease as we dynamically update the community detection module, which proves the community detection can dynamically capture hard samples and the result is helpful for the detection.

### 5.4 Ablation Study

As **CACL** framework can improve the performances of all three backbones, we conduct ablation study to analysis the effects of different model components as well as data structure. We use HGT as backbone and present the results in Table 3:



- To study the impact graph augmentation methods, we remove each method individually. Masking node feature shifting results in the least decrease in accuracy compared to other methods. Removing link prediction leads to a sharp decrease in performance, indicating the importance of finding potential social relationships.
- To evaluate the effectiveness of leveraging multiple user information modalities, we mask users' metadata, texts and remove the graph's heterogeneity separately. Compared with the full model, any absence of information from the social networks results in decreased performance, which reveals the importance of all these kinds of user information for bot detection.
- To investigate the importance of community-aware hard samples mining methods, we mask the community-aware module and process the origin graph with GNNs directly. Final results show a decrease in performance, which proves the significance of community-aware module.

## 6 Conclusion

Social media bot detection is an application problem that integrates various fields. To leverage the rich community structure information in social networks and address over-smoothness problem in GNNs model, we propose the **CACL** framework, which utilizes community-aware module to mine hard positive and negative samples at graph level, and introduces graph contrastive learning to handle hard samples in node level. **CACL** also employs three efficient graph augmentation methods which enhance GNNs model's robustness and generalization ability. We conduct extensive experiments on three comprehensive benchmarks which demonstrate the effectiveness of our framework.

## Limitations

In this paper, we mainly focus on constructing a community-aware graph contrastive learning framework. Despite our best efforts, this paper may still have some limitations. We evaluate our method in three benchmarks, including the most comprehensive dataset Twibot-22 in Twitter bot detection area. However, [Hays et al. \(2023\)](#) still point out that current datasets still have certain limitations in collecting and labeling that affect the evaluation of methods. If we find several communities in real world and record the changes of them during the

training process, we can simply reveal the effectiveness of our proposed framework. However, due to our framework is dynamic updated, community structure is dynamic and hard to record.

Large language models exhibit a high level of comprehension ability for graph information. Effectively integrating LLM technology can enhance the extraction of user information, which contributes to the identification of robot identities. For example, [Liu et al. \(2023\)](#) use mixture of experts to improve the performance and [Tang et al. \(2023\)](#) employ GPT to deal with graph data. Despite their excellent performances, we focus on enhancing the performance of GNNs models, so we do not delve into these aspects too much in this paper.

## Ethics Statement

Our research in this paper follows ethical guidelines and principles. All experiments and data usage are conducted in accordance with the relevant institutional and national guidelines and regulations. Despite the datasets we use contain a large amount of private information existing in social media, we ensure the privacy and confidentiality of any personal or sensitive information used and obtained during the study. Our work aims to contribute positively to online environment, while respecting the dignity, rights, and well-being of individuals and communities. We acknowledge and address any potential biases or ethical considerations in our research.

## Acknowledgements

This research is supported by the National Natural Science Foundation of China (No.62106105), the CCF-Baidu Open Fund (No.CCF-Baidu202307), the CCF-Zhipu AI Large Model Fund (No.CCF-Zhipu202315), the Scientific Research Starting Foundation of Nanjing University of Aeronautics and Astronautics (No.YQR21022), the High Performance Computing Platform of Nanjing University of Aeronautics and Astronautics, and the foundation of National College Students' innovation and entrepreneurship training program (No.202310287085Z).

## References

2004. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113.

- 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 of The 2019 World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 148–153. ACM.
- Jonathon M Berger and Jonathon Morgan. 2015. The isis twitter census: Defining and describing the population of isis supporters on twitter.
- Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008.
- Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. 2020. [Measuring and relieving the over-smoothing problem for graph neural networks from the topological view](#). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 3438–3445. AAAI Press.
- Hocine Cherifi, Gergely Palla, Boleslaw K. Szymanski, and Xiaoyan Lu. 2019. [On community structure in complex networks: challenges and opportunities](#). *Appl. Netw. Sci.*, 4(1):117.
- Stefano Cresci. 2020. [A decade of social bot detection](#). *Commun. 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](#). *Decis. Support Syst.*, 80:56–71.
- Stefano Cresci, Roberto Di Pietro, Angelo Spognardi, Maurizio Tesconi, and Marinella Petrocchi. 2023. [Demystifying misconceptions in social bots research](#). *CoRR*, abs/2303.17251.
- Ashkan Dehghan, Kinga Siuta, Agata Skorupka, Akshat Dubey, Andrei Betlen, David Miller, Wei Xu, Bogumil Kaminski, and Pawel Pralat. 2023. [Detecting bots in social-networks using node and structural embeddings](#). *J. Big Data*, 10(1):119.
- Shangbin Feng, Zhaoxuan Tan, Rui Li, and Minnan Luo. 2022a. [Heterogeneity-aware twitter bot detection with relational graph transformers](#). In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 3977–3985. AAAI Press.
- Shangbin Feng, Zhaoxuan Tan, Herun Wan, Ningnan Wang, Zilong Chen, Binchi Zhang, Qinghua Zheng, Wenqian Zhang, Zhenyu Lei, Shujie Yang, Xinsun Feng, Qingyue Zhang, Hongrui Wang, Yuhan Liu, Yuyang Bai, Heng Wang, Zijian Cai, Yanbo Wang, Lijing Zheng, Zihan Ma, Jundong Li, and Minnan Luo. 2022b. [Twibot-22: Towards graph-based twitter bot detection](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Shangbin Feng, Herun Wan, Ningnan Wang, Jundong Li, and Minnan Luo. 2021a. [SATAR: A self-supervised approach to twitter account representation learning and its application in bot detection](#). In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 3808–3817. ACM.
- Shangbin Feng, Herun Wan, Ningnan Wang, Jundong Li, and Minnan Luo. 2021b. [Twibot-20: A comprehensive twitter bot detection benchmark](#). In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 4485–4494. ACM.
- Shangbin Feng, Herun Wan, Ningnan Wang, and Minnan Luo. 2021c. [Botrgcn: Twitter bot detection with relational graph convolutional networks](#). In *ASONAM '21: International Conference on Advances in Social Networks Analysis and Mining, Virtual Event, The Netherlands, November 8 - 11, 2021*, pages 236–239. ACM.
- Emilio Ferrara. 2017. [Disinformation and social bot operations in the run up to the 2017 french presidential election](#). *First Monday*, 22(8).
- Santo Fortunato. 2010. Community detection in graphs. *Physics reports*, 486(3-5):75–174.
- Qinglang Guo, Haiyong Xie, Yangyang Li, Wen Ma, and Chao Zhang. 2022. [Social bots detection via fusing BERT and graph convolutional networks](#). *Symmetry*, 14(1):30.
- William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. [Inductive representation learning on large graphs](#). In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1024–1034.
- Kadhim Hayawi, Sujith Samuel Mathew, Neethu Venugopal, Mohammad M. Masud, and Pin-Han Ho. 2022. [Deeprobot: a hybrid deep neural network model for social bot detection based on user profile data](#). *Soc. Netw. Anal. Min.*, 12(1):43.
- Chris Hays, Zachary Schutzman, Manish Raghavan, Erin Walk, and Philipp Zimmer. 2023. [Simplistic collection and labeling practices limit the utility of benchmark datasets for twitter bot detection](#). In *Proceedings of the ACM Web Conference 2023, WWW*

- 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023, pages 3660–3669. ACM.
- Dongxiao He, Yue Song, Di Jin, Zhiyong Feng, Binbin Zhang, Zhizhi Yu, and Weixiong Zhang. 2020. [Community-centric graph convolutional network for unsupervised community detection](#). In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3515–3521. ijcai.org.
- Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. [Heterogeneous graph transformer](#). In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 2704–2710. ACM / IW3C2.
- Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschiot, Ce Liu, and Dilip Krishnan. 2020. [Supervised contrastive learning](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Thomas N. Kipf and Max Welling. 2016. [Semi-supervised classification with graph convolutional networks](#). *CoRR*, abs/1609.02907.
- Sneha Kudugunta and Emilio Ferrara. 2018. [Deep neural networks for bot detection](#). *Inf. Sci.*, 467:312–322.
- Bolian Li, Baoyu Jing, and Hanghang Tong. 2022. [Graph communal contrastive learning](#). In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 1203–1213. ACM.
- 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](#). *CoRR*, abs/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, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 485–495. ACM.
- 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 *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pages 1150–1160. ACM.
- Jiaqi Ma, Junwei Deng, and Qiaozhu Mei. 2021. [Subgroup generalization and fairness of graph neural networks](#). In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 1048–1061.
- George A. Miller. 1995. [Wordnet: A lexical database for english](#). *Commun. ACM*, 38(11):39–41.
- Nam P Nguyen, Thang N Dinh, Yilin Shen, and My T Thai. 2014. [Dynamic social community detection and its applications](#). *PloS one*, 9(4):e91431.
- Mariam Orabi, Djedjiga Mouheb, Zaher Al Aghbari, and Ibrahim Kamel. 2020. [Detection of bots in social media: A systematic review](#). *Inf. Process. Manag.*, 57(4):102250.
- Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. [The pagerank citation ranking: Bringing order to the web](#). *the web conference*, 98.
- J. Ross Quinlan. 1986. [Induction of decision trees](#). *Mach. Learn.*, 1(1):81–106.
- Guillaume Salha-Galvan, Johannes F. Lutzeyer, George Dasoulas, Romain Hennequin, and Michalis Vazirgiannis. 2022. [Modularity-aware graph autoencoders for joint community detection and link prediction](#). *Neural Networks*, 153:474–495.
- Zhaoxuan Tan, Shangbin Feng, Melanie Sclar, Herun Wan, Minnan Luo, Yejin Choi, and Yulia Tsvetkov. 2023. [Botpercent: Estimating twitter bot populations from groups to crowds](#). *CoRR*, abs/2302.00381.
- Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. 2023. [Graphgpt: Graph instruction tuning for large language models](#). *CoRR*, abs/2310.13023.
- Cunchao Tu, Xiangkai Zeng, Hao Wang, Zhengyan Zhang, Zhiyuan Liu, Maosong Sun, Bo Zhang, and Leyu Lin. 2019. [A unified framework for community detection and network representation learning](#). *IEEE Trans. Knowl. Data Eng.*, 31(6):1051–1065.
- Onur Varol, Emilio Ferrara, Clayton A. Davis, Filippo Menczer, and Alessandro Flammini. 2017. [Online human-bot interactions: Detection, estimation, and characterization](#). In *Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM 2017, Montréal, Québec, Canada, May 15-18, 2017*, pages 280–289. AAAI Press.
- Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. [Graph attention networks](#). *CoRR*, abs/1710.10903.
- Zhen Wang, Chunyu Wang, Chao Gao, Xuelong Li, and Xianghua Li. 2020. [An evolutionary autoencoder for dynamic community detection](#). *Sci. China Inf. Sci.*, 63(11).
- Qi Wu, Yingguang Yang, Buyun He, Hao Liu, Xiang Wang, Yong Liao, Renyu Yang, and Pengyuan Zhou. 2023. [Heterophily-aware social bot detection with supervised contrastive learning](#). *CoRR*, abs/2306.07478.

- Yuhao Wu, Yuzhou Fang, Shuaikang Shang, Jing Jin, Lai Wei, and Haizhou Wang. 2021. [A novel framework for detecting social bots with deep neural networks and active learning](#). *Knowl. Based Syst.*, 211:106525.
- Chao Yang, Robert Chandler Harkreader, and Guofei Gu. 2013. [Empirical evaluation and new design for fighting evolving twitter spammers](#). *IEEE Trans. Inf. Forensics Secur.*, 8(8):1280–1293.
- Kai-Cheng Yang, Onur Varol, Pik-Mai Hui, and Filippo Menczer. 2020. [Scalable and generalizable social bot detection through data selection](#). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 1096–1103. AAAI Press.
- 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 Trans. Web*, 17(3):15:1–15:31.
- Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. [Graph contrastive learning with augmentations](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Jun Zhao, Xudong Liu, Qiben Yan, Bo Li, Minglai Shao, and Hao Peng. 2020. [Multi-attributed heterogeneous graph convolutional network for bot detection](#). *Inf. Sci.*, 537:380–393.
- Kaixiong Zhou, Xiao Huang, Yuening Li, Daochen Zha, Rui Chen, and Xia Hu. 2020. [Towards deeper graph neural networks with differentiable group normalization](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Ming Zhou, Dan Zhang, Yuandong Wang, Yangli-Ao Geng, and Jie Tang. 2023a. [Detecting social bot on the fly using contrastive learning](#). In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, pages 4995–5001. ACM.
- Yuchen Zhou, Yanan Cao, Yongchao Liu, Yanmin Shang, Peng Zhang, Zheng Lin, Yun Yue, Baokun Wang, Xing Fu, and Weiqiang Wang. 2023b. [Multi-aspect heterogeneous graph augmentation](#). In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pages 39–48. ACM.
- Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. [Deep graph contrastive representation learning](#). *CoRR*, abs/2006.04131.
- Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. [Graph contrastive learning with adaptive augmentation](#). In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 2069–2080. ACM / IW3C2.