# Identifying Bots on Social Media through Coordinated Group Perception

*Boyu Qiao*[⋆†], *Kun Li*[⋆⋄], *Wei Zhou*[⋆†], *Shilong Li*[⋆†], *Qianqian Lu*[⋆], *Songlin Hu*[⋆]

[⋆]Institute of Information Engineering, Chinese Academy of Sciences
[†]School of Cyber Security, University of Chinese Academy of Sciences
{qiaoboyu, likun2, zhouwei, lishilong, luqianqian, husonglin}@iie.ac.cn

***Abstract*—Identifying bots on social media has become a crucial and challenging task for regulating online discourse. Existing detection methods primarily focus on individual account-level information, identifying potential threats by detecting inconsistencies between genuine humans and anomalous bots in personal profiles, textual content, and social relationships. However, these approaches generally overlook the coordinated behavior characteristics inherent in groups of bot accounts. To address this research gap, we propose a novel Bot detection network based on Coordinated Group Perception (BotCGP), which enhances bot identification performance by uncovering the collective coordinated features among bot groups. Specifically, our method jointly models account profiles, textual content, and social relationships using a Student's t-distribution kernel function and a differentiable modularity function to capture potential coordinated characteristics. Experimental results demonstrate that BotCGP significantly outperforms existing methods in bot detection across three real-world X/Twitter datasets. Our code is available at https://github.com/QQQQQQBY/BotCGP.**

***Index Terms*—Coordinated Characteristics, Account Groups, Differentiable Modularity**

## I. Introduction

The presence of social bots poses risks to online social networks (OSNs) security, becoming significant influencers in shaping public opinion [1], disrupting information dissemination [2], and restricting freedom of speech rights [3]. For example, these bots actively participate in significant events such as the 2018 [4] and 2020 [5] U.S. presidential elections, the COVID-19 global pandemic [6], [7], and online cryptocurrency discussions [8]. Consequently, designing effective methods for bot identification is of great significance in maintaining the integrity and normal functioning of OSNs.

Previous detection researches focus on utilizing individual account-level user information to identify threats by recognizing inconsistencies between genuine human users and anomalous bots in personal profiles, posted content, and social relationships. For instance, Yang et al. [9] use user profile features, their derived characteristics, and machine learning methods to assess bot activity. Wei et al. [10] and Kudugunta et al. [11] employ Bi-LSTM [12] to analyze the content and contextual semantic features of user posts for detection. Additionally, Feng et al. [13], [14] and Li et al. [15] construct graph structures based on social relationships and respectively utilize graph neural networks (GNN) and graph augmentation methods to enhance bot detection performance.

However, existing bot detection methods generally ignore the coordinated behavioral information hidden within bot account groups. Specifically, to save costs and increase efficiency, bot operators often create and control large numbers of accounts simultaneously, resulting in noticeable similarities in profile attributes, such as creation dates and post counts. Furthermore, these bot groups exhibit coordinated behavior in their textual content, frequently posting about the same topics to amplify specific narratives [16]–[18]. For example, they engage in coordinated postings to manipulate stock prices [19], [20] or cryptocurrency prices [8]. Additionally, the social interactions of bot accounts are usually driven by specific objectives and programmed algorithms, engaging in coordinated actions such as follows, mentions, or replies through bulk operations to boost the popularity of target users or topics [21]–[23]. In contrast, human users tend to have personalized profiles with subjective and diverse content, and their interactions are based on shared interests or real-world connections [3], [24]. Their behavior appears more natural and fluid, lacking the obvious coordinated patterns seen in bot activity.

Motivated by the limitations of the existing methods, in this paper, we propose an end-to-end Bot detection network based on Coordinated Group Perception, namely BotCGP. Our model consists of two main components: a coordinated group perception module and a coordinated feature-enhanced detection network. In the coordinated group perception module, we jointly model account profiles, textual content, and social relationship information based on a Student's t-distribution kernel function and a differentiable modularity function, to capture latent coordinated group features. In the coordinated feature-enhanced detection network, we integrate the initial attribute features of accounts with the extracted coordinated group features and employ Relational Graph Convolutional Networks (RGCN) [25] to aggregate the feature information of neighboring nodes, thereby obtaining more comprehensive user representations. We then perform node classification to identify bot accounts. Experimental results on three real-world datasets validate the superior performance of BotCGP compared to existing bot detection methods.

## II. Methodology

The BotCGP framework is illustrated in Figure 1. Our model comprises two main parts: the coordinated group perception module and the coordinated feature-enhanced detection network.

### A. Problem Formulation

Within the user set $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_n\}$, where $n$ is the number of accounts, user information is comprised of profiles, textual content, and social relationships. Profiles include value-type attributes $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n\}$, such as follower counts, and boolean-type attributes $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n\}$, such as whether the user shares their location. Textual content denoted by $\mathbf{T} = \{\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_n\}$ is encoded by pre-trained language model (PLM) [26], [27]. Social relationships construct the topology graph $\mathcal{G} = \{\mathbf{V}, \mathbf{E}|_{r=0}^{\mathbf{R}}, \mathbf{Y}\}$, where $\mathbf{E}$ represents the edges set connecting users, $\mathbf{R}$ denotes the number of edge types, and $\mathbf{Y}$ indicates user labels. The objective of bot identification is to find a function $f : f(\mathbf{u}_i) \rightarrow \hat{\mathbf{y}}_i \in \{0, 1\}$, effectively distinguishing between genuine users and anomalous bots.

[⋄] Corresponding Author: Kun Li, likun2@iie.ac.cn

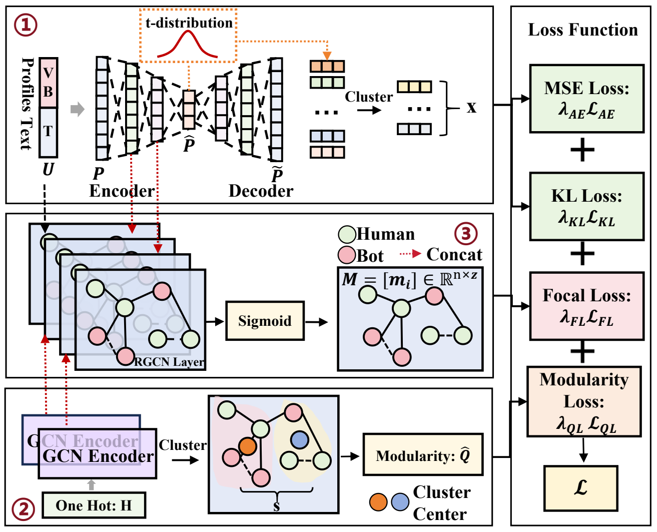

①
t-distribution
Profiles Text
V
B
T
U
P
$\hat{P}$
Encoder
Decoder
$\tilde{P}$
Cluster
x
Loss Function
MSE Loss: $\lambda_{AE}\mathcal{L}_{AE}$
KL Loss: $\lambda_{KL}\mathcal{L}_{KL}$
Focal Loss: $\lambda_{FL}\mathcal{L}_{FL}$
Modularity Loss: $\lambda_{QL}\mathcal{L}_{QL}$
$\mathcal{L}$
Human
Bot
③
Concat
$M = [m_i] \in \mathbb{R}^{n\times z}$
Sigmoid
RGCN Layer
GCN Encoder
Cluster
One Hot: H
②
s
Modularity: $\hat{Q}$
Cluster Center

Fig. 1. The overall structure of **BotCGP**.

## *B. Coordinated Group Perception Module*

In the coordinated group perception module, we extract coordinated group features from user profiles, textual content, and social relationship networks.

**Coordinated group features of user profiles and textual content.** We utilize autoencoders [28] and K-means to extract coordinated group features of user profiles and textual content. Then, we train and optimize this component using a loss function based on the Student's t-distribution kernel.

On the encoder side, the user's value-type attributes $V$, boolean-type attributes $B$, and posted textual content attributes $T$ are first concatenated to form the input, $\mathbf{U} = Concat(\mathbf{V};\mathbf{B};\mathbf{T})$. This input is then processed by Multi-Layer Perceptrons (MLPs) to represent the user information as dense vectors:

$$\mathbf{P}^{(l+1)} = \sigma(MLP(\mathbf{P}^{(l)})), \tag{1}$$

where $\sigma$ is the activation function. $\mathbf{P}^{(l+1)}$ is the output feature matrix of layer $l$ at the encoder side, with $\mathbf{P}^{(0)} = \mathbf{U}$. The output of the last layer denoted as $\hat{\mathbf{P}} = \{\hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2, ..., \hat{\mathbf{p}}_n\}$. Similarly, the decoder is also modeled as MLPs:

$$\mathbf{P}'^{(l+1)} = \sigma(MLP(\mathbf{P}'^{(l)})), \tag{2}$$

where $\mathbf{P}'^{(l+1)}$ denotes the output of $l$-th decoder layer, with $\mathbf{P}'^{(0)} = \hat{\mathbf{P}}$. The final layer output is represented as $\tilde{\mathbf{P}} = \{\tilde{\mathbf{p}}_1, \tilde{\mathbf{p}}_2, ..., \tilde{\mathbf{p}}_n\}$.

After obtaining the encoded representations $\hat{\mathbf{P}}$, we input $\hat{\mathbf{P}}$ into the K-means to cluster the potential coordinated groups. The representations $\hat{\mathbf{P}}$ are initially clustered into $x$ groups, with the trainable cluster center vectors denoted as $\mathbf{K} = \{\mathbf{k}_1, \mathbf{k}_2, ..., \mathbf{k}_x\}$.

The purpose of clustering is to minimize the distances within the same coordinated groups while maximizing the distances between different groups. For the $i$-th encoded vector and the $j$-th cluster, inspired by the clustering optimization methods in [29], we use the Student's t-distribution as the kernel to measure the similarity between $\hat{\mathbf{P}}$ and $\mathbf{K}$:

$$\mathbf{o}_{ij} = \frac{(1 + ||\hat{\mathbf{p}}_i - \mathbf{k}_j||^2/\tau)^{-\frac{\tau+1}{2}}}{\sum_j^x (1 + ||\hat{\mathbf{p}}_i - \mathbf{k}_j||^2/\tau)^{(-\frac{\tau+1}{2})}}, \tag{3}$$

where $\mathbf{O} = [\mathbf{o}_{ij}] \in \mathbb{R}^{n\times x}$ represents the similarity between the $i$-th user embedding and the $j$-th cluster center, and $\tau$ is the degrees of freedom of the t-distribution. To further enhance the cohesion of clustering, we compute the target similarity distributions with the following equation:

$$\hat{\mathbf{o}}_{ij} = \frac{\mathbf{o}_{ij}^2 / \sum_i^n \mathbf{o}_{ij}}{\sum_j^x \mathbf{o}_{ij}^2 / \sum_i^n \mathbf{o}_{ij}}. \tag{4}$$

We define $\hat{\mathbf{O}} = [\hat{\mathbf{o}}_{ij}] \in \mathbb{R}^{x\times n}$ as the normalized matrix of $\mathbf{O}$. Squaring the similarity values helps to magnify the similarities within similar groups while diminishing those between dissimilar groups.

Mean Square Error (MSE) and Kullback-Leibler (KL) divergence [30] are utilized as the auto-encoder and K-means clustering loss functions:

$$\begin{aligned} \mathcal{L}_{AE} &= \frac{1}{n}\sum_{i=1}^{n} ||\mathbf{p}_i - \tilde{\mathbf{p}}_i||_2^2, \\ \mathcal{L}_{KL} &= KL(\mathbf{O}||\hat{\mathbf{O}}) \\ &= \frac{1}{n\times x}\sum_i^n\sum_j^x \mathbf{o}_{ij} log\frac{\mathbf{o}_{ij}}{\hat{\mathbf{o}}_{ij}}, \end{aligned} \tag{5}$$

where $KL(\cdot||\cdot)$ denotes the distance between two distributions. By minimizing $\mathcal{L}_{KL}$, the coordinated group features are integrated into user representations, fostering the ability to distinguish between humans and bots.

**Coordinated group features of social relationships networks.** We employ Graph Convolutional Networks (GCN) [31] and clustering to extract coordinated group features at the structural level of social networks. Then, we train and optimize this component using differentiable modularity.

We first define $\mathbf{A}$ as the adjacency matrix of the social graph $\mathcal{G}$ and initialize the node representations $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_u\}$ using one-hot vectors. Next, we utilize a GCN encoder to learn the structural features:

$$\hat{\mathbf{H}}^{(l+1)} = \sigma(\hat{\mathbf{H}}^{(l)}\mathbf{W}_g^{(l)}\hat{\mathbf{A}} + \mathbf{b}_g^{(l)}), \tag{6}$$

where $\hat{\mathbf{H}}^{(0)} = \mathbf{H}$, $\hat{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$, and $\mathbf{D}$ is the degree matrix of $\mathcal{G}$. $\mathbf{W}_g^{(l)}$ and $\mathbf{b}_g^{(l)}$ are the weight matrix and bias of the $l$-th GCN layer.

After obtaining the updated node representations $\hat{\mathbf{H}}$, we process the embeddings into $s$ clusters, with the cluster center vectors denoted as $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_s\}$. We measure the similarity of the $i$-th node embeddings to the $j$-th cluster centers using the matrix product and the normalization:

$$\mathbf{f}_{ij} = \frac{exp(\hat{\mathbf{h}}_i\mathbf{c}_j^T)}{\sum_j^s exp(\hat{\mathbf{h}}_i\mathbf{c}_j^T)}, \tag{7}$$

where $\mathbf{f}_{ij}$ denotes the probability of node $i$ being assigned to the $j$-th cluster, for all $j$, $\sum_j^s \mathbf{f}_{ij} = 1$ and $\mathbf{F} = [\mathbf{f}_{ij}] \in \mathbb{R}^{n\times s}$. We employ maximizing the modularity [32], to train the current module:

$$Q = \frac{1}{2m}\sum_a^n\sum_c^n(\mathbf{A}_{ac} - \frac{d_a d_c}{2m})\delta(l_a, l_c), \tag{8}$$

where $\mathbf{A}_{ac}$ represents the value of the adjacency matrix, $d_a, d_c$ represents the degree of nodes $a$, and $c$, $m$ denotes the edge counts. $l_a$ and $l_c$ denote the coordinated groups to which nodes $a$ and $c$ belong. The $\delta(l_a, l_c)$ is defined as:

$$\begin{cases} \delta(l_a, l_c) = 1, & l_a = l_c \\ \delta(l_a, l_c) = 0, & l_a! = l_c. \end{cases} \tag{9}$$

Equations (8) and (9) show that the traditional modularity calculation function is non-differentiable, making it incompatible with the gradient-based optimization required for model training. Inspired by Wilder et al. [37], we use a differentiable modularity function

TABLE I

MAIN RESULTS. WE RUN EACH METHOD FIVE TIMES AND REPORT THE AVERAGE VALUE AND STANDARD DEVIATION. UNDERLINE INDICATES SUBOPTIMAL, AND "-" INDICATES THAT THE METHOD IS NOT SCALABLE TO MGTAB-22. W/O ①, W/O ② AND W/O ① & ② REPRESENT THE REMOVAL OF COMPONENTS ①, ②, AND BOTH ① & ② FROM FIGURE 1.

| Method | Cresci-15 | | Twibot-20 | | MGTAB-22 | |
|---|---|---|---|---|---|---|
| | Accuracy(%) | F1-score(%) | Accuracy(%) | F1-score(%) | Accuracy(%) | F1-score(%) |
| SGBOT [9] | 77.08 (±0.21) | 77.91 (±0.11) | 81.64 (±0.46) | 84.89 (±0.42) | - | - |
| Wei et al. [10] | 96.18 (±1.54) | 82.65 (±2.21) | 70.23 (±0.10) | 53.61 (±0.10) | - | - |
| BotRGCN [13] | 96.52 (±0.71) | 97.30 (±0.53) | 83.27 (±0.57) | 85.26 (±0.38) | 89.27 (±1.27) | 86.01 (±2.11) |
| HGT [33] | 96.10 (±0.21) | 96.93 (±0.31) | 85.62 (±0.25) | 87.21 (±0.27) | 89.78 (±1.48) | 87.02 (±2.03) |
| RGT [14] | 97.15 (±0.32) | 97.78 (±0.24) | 86.57 (±0.41) | 88.01 (±0.41) | 90.06 (±0.92) | 87.06 (±1.34) |
| BothH [15] | <u>98.80 (±0.23)</u> | <u>98.87 (±0.21)</u> | <u>87.74 (±0.51)</u> | 89.28 (±0.38) | 90.29 (±1.47) | 87.54 (±2.01) |
| BECE [34] | 98.31 (±0.62) | 98.67 (±0.48) | 87.50 (±0.38) | <u>89.54 (±0.28)</u> | <u>90.96 (±1.84)</u> | <u>88.21 (±2.45)</u> |
| BIC [35] | 98.35 (±0.24) | 98.71 (±0.18) | 87.61 (±0.21) | 89.13 (±0.15) | - | - |
| BotMoE [36] | 98.50 (±0.00) | 98.82 (±0.00) | 87.54 (±0.22) | 89.22 (±0.31) | 87.45(±0.87) | 78.23 (±1.31) |
| **W/O ①** | 99.12 (±0.14) | 98.38 (±0.12) | 88.18 (±0.44) | 88.23 (±0.68) | 90.20 (±0.80) | 87.61 (±0.69) |
| **W/O ②** | 98.98 (±0.16) | 98.23 (±0.23) | 87.86 (±0.58) | 88.13 (±0.54) | 90.33 (±0.74) | 87.80 (±0.66) |
| **W/O ① & ②** | 98.71 (±0.11) | 97.54 (±0.14) | 87.28 (±0.32) | 87.44 (±0.29) | 90.15 (±1.00) | 87.19 (±0.99) |
| **BotCGP** | **99.43 (±0.22)** | **99.71 (±0.23)** | **88.79 (±0.28)** | **89.98 (±0.34)** | **91.27 (±0.87)** | **88.69 (±0.99)** |

to optimize the coordinated group perception of social relationships. The formula for the differentiable modularity function is:

$$\hat{Q} = \frac{1}{2m}\sum_{a}^{n}\sum_{c}^{n}(\mathbf{A}_{ac} - \frac{d_a d_c}{2m})\mathbf{f}_a\mathbf{f}_c^T, \tag{10}$$

where $\mathbf{f}_a = [\mathbf{f}_{a1}, \mathbf{f}_{a2}, ..., \mathbf{f}_{as}]$ and $\mathbf{f}_c = [\mathbf{f}_{c1}, \mathbf{f}_{c2}, ..., \mathbf{f}_{cs}]$. $\mathbf{f}_a$ and $\mathbf{f}_c$ represent the probabilities that nodes $a$ and $c$ belong to each of the $s$ different groups. $\mathbf{f}_a\mathbf{f}_c^T$ denotes the probability that nodes $a$ and $c$ belong to the same group, with values range $(0, 1)$. We utilize $\mathbf{f}_a\mathbf{f}_c^T$ to replace the discrete part $\delta(l_a, l_c)$ in the Equation (8), thereby relaxing the binary quantities to fractional values within the range $(0, 1)$, and making the modularity function differentiable.

We use the differentiable modularity as a loss function for optimization:

$$\mathcal{L}_{QL} = -\hat{Q}. \tag{11}$$

The negative sign "$-$" is used because a higher modularity value signifies better-coordinated group segmentation. Therefore, minimizing the $\mathcal{L}_{QL}$ loss function aims to maximize the modularity value.

### C. Coordinated Feature-enhanced Detection Network

We utilize RGCN to jointly learn users' coordinated features, attribute features, and neighbor features, and then perform node classification for bot detection.

Similar to previous work [14], [36], we first represent user information using the textual content $\mathbf{T}$, profiles ($\mathbf{V}$ and $\mathbf{B}$). To create user node representations $\tilde{\mathbf{U}}$, we concatenate the outputs of MLPs applied to each feature type:

$$\tilde{\mathbf{u}}_i = Concat(MLP(\mathbf{v}_i), MLP(\mathbf{b}_i), MLP(\mathbf{t}_i)), \tag{12}$$

we then use RGCN to capture and aggregate neighbor node information and coordinated features above:

$$\begin{aligned}\hat{\mathbf{u}}_i^{(l)} &= \sigma(Concat(\tilde{\mathbf{u}}_i^{(l)}, \hat{\mathbf{h}}_i^{(l)}, \mathbf{p}_i^{(l)})), \\ \hat{\mathbf{u}}_i^{(l+1)} &= \mathcal{COM}(\hat{\mathbf{u}}_i^{(l)}, \mathcal{AGG}(\hat{\mathbf{u}}_j^{(l)}, \forall j \in \mathcal{N}(i))),\end{aligned} \tag{13}$$

where $\mathcal{N}(i)$ denotes the neighboring nodes of node $i$, $\mathcal{AGG}$ denotes the aggregation function, and $\mathcal{COM}$ denotes the update function after aggregating the information of neighboring nodes. $\hat{\mathbf{h}}_i^{(l)}$ and $\mathbf{p}_i^{(l)}$ represent the potential coordinated group features of user $i$.

Finally, the comprehensive coordinated group affiliation features $\mathbf{M}$ are obtained through the Sigmoid activation function:

$$\mathbf{m}_i = Sigmoid(\hat{\mathbf{u}}_i^{(L)}), \tag{14}$$

where $L$ denotes the RGCN layer number. The matrix $\mathbf{M} = [\mathbf{m}_i] \in \mathbb{R}^{n \times z}$ represents the coordinated group affiliation matrix, with the dimension $z$ corresponding to the number of final coordinated groups. Each value in $\mathbf{M}$ indicates the probability of a user belonging to different coordinated groups.

We use focal loss to jointly optimize various features:

$$\begin{aligned}\hat{\mathbf{Y}} &= Softmax(MLP(\mathbf{M})), \\ \mathcal{L}_{FL} &= -\alpha(1 - \hat{\mathbf{Y}})^{\gamma} log(\hat{\mathbf{Y}}),\end{aligned} \tag{15}$$

where $\hat{\mathbf{Y}}$ represents the predicted probabilities of the user being bots, $\alpha$ and $\gamma$ are hyperparameters. $\alpha$ helps to mitigate the imbalance between positive and negative samples, while $\gamma$ adjusts the weight of easy and hard samples.

We train BotCGP using the following loss function:

$$\mathcal{L} = \lambda_{AE}\mathcal{L}_{AE} + \lambda_{KL}\mathcal{L}_{KL} + \lambda_{QL}\mathcal{L}_{QL} + \lambda_{FL}\mathcal{L}_{FL}, \tag{16}$$

where $\lambda_{AE}, \lambda_{KL}, \lambda_{QL}, \lambda_{FL}$ are adjustable parameters.

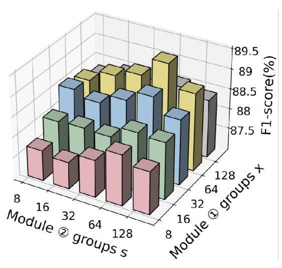
## III. EXPERIMENTS

### A. Experiment Setup

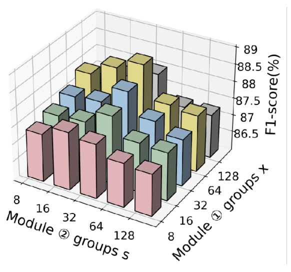
**Dataset:** Following the previous work [34], [38], we select three real-world X/Twitter datasets for further experimental exploration: Cresci-15 [39], Twibot-20 [40] and MGTAB-22 [41].

**Baseline Methods:** We compare detection methods based on profiles and textual content, such as SGBOT [9] and Wei et al. [10]. We also evaluate advanced methods that leverage social relationship networks, including BotRGCN [13], RGT [14], HGT [33], BothH [15] and BECE [34]. Additionally, we compare with the detection method BIC [35], which is based on user text and structural interactions. Furthermore, we analyze BotMoE [36], which uses an expert network to assign users to different communities for bot detection. Although BotMoE utilizes the concept of group division, it does not effectively capture coordinated features, grouping bots and humans into the same communities, thus limiting its detection performance.

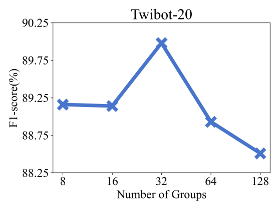
**Implementation Details**: We utilize Adam as the optimizer, with a batch size of 1,024 and a learning rate of 1e-4. Model training extends up to 200 epochs. To ensure direct comparability with earlier research, we adhere to the same splits provided in the benchmark, allocating 70% as the training set, 20% to the validation set, and 10% to the test set. Experiments are conducted on a clustered device with four Tesla V100 GPUs and 32 GB of RAM.



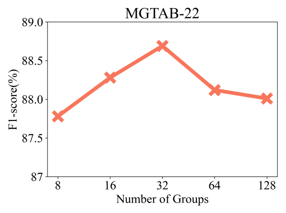
(a) Twibot-20 parameters: $x, s$



(b) MGTAB-22 parameters: $x, s$

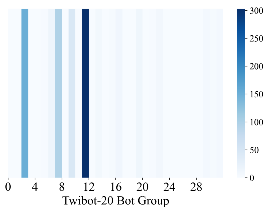


(c) Twibot-20 parameters: $z$

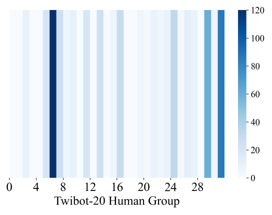


(d) MGTAB-22 parameters: $z$

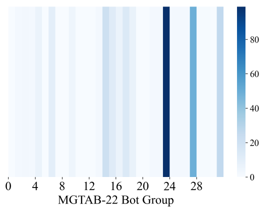
Fig. 2. Parameter analysis results in BotCGP



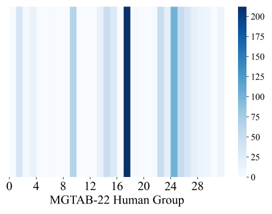
(a) Twibot-20 bot group



(b) Twibot-20 human group



(c) MGTAB-22 bot group



(d) MGTAB-22 human group

Fig. 3. Heatmap of the group affiliation matrix $M$.

## B. Main Results

We perform experiments on the three X/Twitter datasets and compare them with recent bot detection methodologies. Our experimental results are presented in Table I.

BotCGP significantly outperforms all baseline methods across three real-world datasets. BotCGP improves accuracy by 0.63%, 1.05%, and 0.31% on the Cresci-15, Twibot-20, and MGTAB-22 datasets compared to the second-best results. F1-scores also increase by 0.84%, 0.44%, and 0.48% for these datasets. Compared to BotMoE [36], which also incorporates group information, BotCGP demonstrates performance improvements across all three datasets. Notably, on the MGTAB-22 dataset, BotCGP shows a significant accuracy increase of 3.82%. These results highlight the effectiveness of BotCGP in bot detection tasks.

To evaluate the impact of different coordinated group features on BotCGP's performance, we conduct ablation studies. In Table I, 'W/O ①', 'W/O ②', and 'W/O ① & ②' represent the removal of components ①, ②, and both ① & ② from Figure 1, along with their respective loss functions. The removal of these components leads to a significant decline in BotCGP's performance. Notably, removing both ① and ② results in the most substantial performance drop, highlighting the importance of coordinated group features embedded in user profiles, textual content, and interaction structures for effective bot identification.

## C. Coordinated Group Study

To determine the optimal number of coordinated groups, we evaluate three coordinated group-related hyperparameters, $x$, $s$, and $z$. We set the number of groups to $\{8, 16, 32, 64, 128\}$ and conduct experiments for each hyperparameter individually. Figure 2 (a) and (b) show the effects of hyperparameters $x$ (as in ① in Figure 1) and $s$ (as in ② in Figure 1) on the performance of BotCGP. It is evident from the figures that as $x$ and $s$ increase, BotCGP's performance initially improves and then decreases. For the Twibot-20 dataset, the optimal parameters are $\{64, 64\}$, while for the MGTAB-22 dataset, the best values are $\{64, 32\}$. After establishing the optimal values for $x$ and $s$, we proceed to examine hyperparameter $z$ (corresponding to ③ in Figure 1), which represents the final number of coordinated groups to which a user belongs. The results in Figure 2 (c) and (d) show that the best performance is achieved when $z$ is set to 32 for both the Twibot-20 and MGTAB-22 datasets. A smaller $z$ may result in incorrectly grouping users with low group similarity, while a larger $z$ could hinder model convergence.

To evaluate whether the learned coordinated group features can effectively divide bots into different coordinated groups, we visualize the final learned feature vectors $\mathbf{M} = [\mathbf{m}_i] \in \mathbb{R}^{n \times z}$, with $z$ set to 32. In $\mathbf{M}$, each user is represented by a $z$-dimensional vector, and their coordinated group affiliation is determined by selecting the highest value from this vector. Figure 3 presents a heatmap of user group affiliations in the Twibot-20 and MGTAB-22 test sets, where each column represents a group. The color bar on the right represents the number of group users, with darker colors corresponding to groups with larger numbers of users. The heatmap reveals that humans and bots predominantly cluster in different groups. Consistent with our hypothesis, bots tend to concentrate in a few specific coordinated clusters, while human groups are more evenly and diversely distributed.

## IV. Conclusion

In this paper, we propose BotCGP, an end-to-end bot detection method based on coordinated group perception. BotCGP leverages a carefully designed perception module that jointly models coordinated group features from user profiles, textual content, and social relationships using the Student's t-distribution kernel function and the differentiable modularity function. Experimental evaluations on three real-world X/Twitter datasets demonstrate the effectiveness of BotCGP and its coordinated group features for bot detection.

## V. Acknowledgements

This work was supported by the National Key Research and Development Program of China (No. 2022YFC3302102).

## References

[1] M. Orabi, D. Mouheb, Z. Al Aghbari, and I. Kamel, "Detection of bots in social media: a systematic review," *Information Processing & Management*, vol. 57, no. 4, p. 102250, 2020.

[2] Z. Ellaky, F. Benabbou, and S. Ouahabi, "Systematic literature review of social media bots detection systems," *Journal of King Saud University-Computer and Information Sciences*, 2023.

[3] S. Cresci, "A decade of social bot detection," *Communications of the ACM*, vol. 63, no. 10, pp. 72–83, 2020.

[4] L. Luceri, A. Deb, A. Badawy, and E. Ferrara, "Red bots do it better: Comparative analysis of social bot partisan behavior," in *Companion proceedings of the 2019 world wide web conference*, pp. 1007–1012, 2019.

[5] G. Guglielmi, "The next-generation bots interfering with the us election," *Nature*, vol. 587, pp. 21 – 21, 2020.

[6] E. Ferrara, "What types of covid-19 conspiracies are populated by twitter bots?," *First Monday*, 2020.

[7] M. Himelein-Wachowiak, S. Giorgi, A. Devoto, M. Rahman, L. Ungar, H. A. Schwartz, D. H. Epstein, L. Leggio, and B. Curtis, "Bots and misinformation spread on social media: Implications for covid-19," *Journal of medical Internet research*, vol. 23, no. 5, p. e26933, 2021.

[8] L. Nizzoli, S. Tardelli, M. Avvenuti, S. Cresci, M. Tesconi, and E. Ferrara, "Charting the landscape of online cryptocurrency manipulation," *IEEE Access*, vol. 8, pp. 113230–113245, 2020.

[9] K.-C. Yang, O. Varol, P.-M. Hui, and F. Menczer, "Scalable and generalizable social bot detection through data selection," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 1096–1103, 2020.

[10] F. Wei and U. T. Nguyen, "Twitter bot detection using bidirectional long short-term memory neural networks and word embeddings," in *2019 First IEEE International conference on trust, privacy and security in intelligent systems and applications (TPS-ISA)*, pp. 101–109, IEEE, 2019.

[11] S. Kudugunta and E. Ferrara, "Deep neural networks for bot detection," *Information Sciences*, vol. 467, pp. 312–322, 2018.

[12] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[13] S. Feng, H. Wan, N. Wang, and M. Luo, "Botrgcn: Twitter bot detection with relational graph convolutional networks," in *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 236–239, 2021.

[14] S. Feng, Z. Tan, R. Li, and M. Luo, "Heterogeneity-aware twitter bot detection with relational graph transformers," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 3977–3985, 2022.

[15] S. Li, B. Qiao, K. Li, Q. Lu, M. Lin, and W. Zhou, "Multi-modal social bot detection: Learning homophilic and heterophilic connections adaptively," in *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 3908–3916, 2023.

[16] T. Marlow, S. Miller, and J. T. Roberts, "Bots and online climate discourses: Twitter discourse on president trump's announcement of us withdrawal from the paris agreement," *Climate Policy*, vol. 21, no. 6, pp. 765–777, 2021.

[17] T. Khaund, B. Kirdemir, N. Agarwal, H. Liu, and F. Morstatter, "Social bots and their coordination during online campaigns: A survey," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 2, pp. 530–545, 2021.

[18] F. B. Keller, D. Schoch, S. Stier, and J. Yang, "Political astroturfing on twitter: How to coordinate a disinformation campaign," *Political communication*, vol. 37, no. 2, pp. 256–280, 2020.

[19] S. Cresci, F. Lillo, D. Regoli, S. Tardelli, and M. Tesconi, "Cashtag piggybacking: Uncovering spam and bot activity in stock microblogs on twitter," *ACM Transactions on the Web (TWEB)*, vol. 13, no. 2, pp. 1–27, 2019.

[20] R. Fan, O. Talavera, and V. Tran, "Social media bots and stock markets," *European Financial Management*, vol. 26, no. 3, pp. 753–777, 2020.

[21] K.-C. Yang, E. Ferrara, and F. Menczer, "Botometer 101: Social bot practicum for computational social scientists," *Journal of computational social science*, vol. 5, no. 2, pp. 1511–1528, 2022.

[22] O. Varol and I. Uluturk, "Journalists on twitter: self-branding, audiences, and involvement of bots," *Journal of Computational Social Science*, vol. 3, no. 1, pp. 83–101, 2020.

[23] C. Shao, G. L. Ciampaglia, O. Varol, K.-C. Yang, A. Flammini, and F. Menczer, "The spread of low-credibility content by social bots," *Nature communications*, vol. 9, no. 1, pp. 1–9, 2018.

[24] B. Qiao, K. Li, W. Zhou, Z. Yan, S. Li, and S. Hu, "Social bot detection based on window strategy," in *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 2201–2206, IEEE, 2023.

[25] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*, pp. 593–607, Springer, 2018.

[26] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[27] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang, "Language-agnostic bert sentence embedding," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 878–891, 2022.

[28] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," *Machine learning for data science handbook: data mining and knowledge discovery handbook*, pp. 353–374, 2023.

[29] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[30] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[31] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[32] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[33] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *Proceedings of the web conference 2020*, pp. 2704–2710, 2020.

[34] B. Qiao, W. Zhou, K. Li, S. Li, and S. Hu, "Dispelling the fake: Social bot detection based on edge confidence evaluation," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[35] Z. Lei, H. Wan, W. Zhang, S. Feng, Z. Chen, J. Li, Q. Zheng, and M. Luo, "BIC: Twitter bot detection with text-graph interaction and semantic consistency," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 10326–10340, 2023.

[36] Y. Liu, Z. Tan, H. Wang, S. Feng, Q. Zheng, and M. Luo, "Botmoe: Twitter bot detection with community-aware mixtures of modal-specific experts," *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023.

[37] B. Wilder, E. Ewing, B. Dilkina, and M. Tambe, "End to end learning and optimization on graphs," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[38] S. Shi, K. Qiao, J. Yang, B. Song, J. Chen, and B. Yan, "Oversampling strategy in feature space for graphs based class-imbalanced bot detection," *arXiv preprint arXiv:2302.06900*, 2023.

[39] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "Fame for sale: Efficient detection of fake twitter followers," *Decision Support Systems*, vol. 80, pp. 56–71, 2015.

[40] S. Feng, H. Wan, N. Wang, J. Li, and M. Luo, "Twibot-20: A comprehensive twitter bot detection benchmark," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 4485–4494, 2021.

[41] S. Shi, K. Qiao, J. Chen, S. Yang, J. Yang, B. Song, L. Wang, and B. Yan, "Mgtab: A multi-relational graph-based twitter account detection benchmark," *arXiv preprint arXiv:2301.01123*, 2023.