BotVA: Combating Social Bots via Variational Feature Augmentation and Adversarial Graph Learning

Longlong Zhang[1,2], **Xi Wang**[3], **Hongyi Nie**[4], **Zeqing Zhang**[5], **Huixiang Zhang**[5], **Hongping Wang**[6], **Yang Liu**[1,2*]

[1]School of Artificial Intelligence, Optics and Electronics, Northwestern Polytechnical University
[2]Shenzhen Research Institute, Northwestern Polytechnical University
[3]Department of Computer Science and Engineering, The Hong Kong University of Science and Technology
[4]School of Mechanical Engineering, Northwestern Polytechnical University
[5]School of Cybersecurity, Northwestern Polytechnical University
[6]School of Intelligent Engineering and Automation, Beijing University of Posts and Telecommunications
{zhanglonglong, hy_nie}@mail.nwpu.edu.cn, vancywangxi@ust.hk, zhanghuixiang@nwpu.edu.cn, {zekchong, hongpingwang105, yangliuyh}@gmail.com

Abstract

Social bots threaten online platforms by spreading disinformation and manipulating public discourse. Graph neural networks have emerged as effective tools for bot detection by modeling user interactions, yet two fundamental challenges limit their practical deployment: severe class imbalance where bots constitute a small minority of users, and camouflaged edges where bots forge deceptive connections to humans to evade detection. Class imbalance causes decision boundaries to shift toward the majority class, while camouflaged edges corrupt neighborhood aggregation through spurious message passing. We present BotVA, a unified framework addressing both challenges through variational feature augmentation and adversarial graph learning. Our approach makes three key contributions: (i) a conditional variational autoencoder that models minority class distributions and synthesizes semantically coherent features, effectively expanding minority support in representation space; (ii) an adversarial training paradigm where a generator simulates camouflage by injecting deceptive edges while a graph transformer discriminator with semantic attention learns to identify and downweight such perturbations; and (iii) a two-stage training strategy that pretrains the variational module before alternating generator-discriminator optimization to ensure stable convergence. Experiments on three benchmarks demonstrate that BotVA achieves state-of-the-art accuracy and F1-score, exhibits strong robustness under camouflage perturbations, and maintains competitive performance with limited supervision.

*Corresponding author.

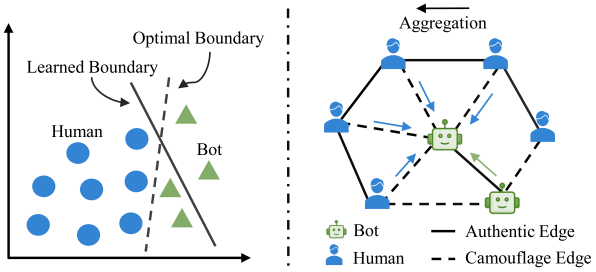

Figure 1: Two fundamental challenges for graph-based social bot detection: class imbalance and camouflaged edges.

1 Introduction

The rapid proliferation of social bots on social media platforms poses systemic risks to information integrity and platform security [Cresci, 2020]. Operated by malicious actors, these automated accounts can strategically disseminate disinformation [Starbird, 2019], manipulate public opinion [Caldarelli *et al.*, 2020], interfere with political processes [Deb *et al.*, 2019], and steadily erode trust in online discourse [Javed *et al.*, 2024]. These harms compound over time and at scale, distorting public communication and increasing content moderation and governance costs for platforms [Shao *et al.*, 2018]. Against this backdrop, developing robust bot detection methods is essential for safeguarding the long-term health of social media ecosystems and preserving a trustworthy online public sphere.

Early work on social bot detection relied on handcrafted features derived from user metadata [Yang *et al.*, 2020], textual content [Kudugunta and Ferrara, 2018], or structural properties of social graphs [Beskow and Carley, 2018], typically coupled with classical classifiers such as random forests [Schnebly and Sengupta, 2019] and support vector machines [Alarifi *et al.*, 2016]. Such feature-based pipelines depend strongly on domain knowledge and have limited capacity to

capture higher-order dependencies across network neighborhoods. To overcome these limitations, recent studies model social media as graphs and employ graph neural networks (GNNs) [Feng *et al.*, 2021c; Feng *et al.*, 2022], which encode node attributes together with topology to represent interaction patterns more effectively. Representative examples include BotDGT [He *et al.*, 2024], which adopts dynamic graphs to model temporal and relational dependencies, thereby improving detection of complex and adaptive bots, and CAGCL [Wei *et al.*, 2025], which leverages latent community discovery and contrastive learning to yield more discriminative representations. Despite these advances, translating graph-based methods into reliable detectors in real-world deployments remains challenging. As illustrated in Figure 1, two fundamental obstacles persist and often co-occur in practice:

Class Imbalance. While curated benchmarks may have varying class ratios, real-world studies [Cresci, 2020; Javed *et al.*, 2024] confirm bots constitute only 8-18% in actual networks. This severe class imbalance biases empirical risk minimization toward the majority class, causing decision boundaries to shift away from optimal separating hyperplanes and reducing recall for minority instances [Johnson and Khoshgoftaar, 2019]. Models trained on such skewed distributions tend to maximize overall accuracy by prioritizing abundant human accounts, resulting in the systematic misclassification of scarce bot accounts.

Camouflaged Edges. Bots often establish deceptive connections to humans to evade detection. These camouflaged edges violate the homophily assumption underlying most GNNs, which expect connected nodes to share similar attributes and labels [Kipf and Welling, 2017]. When bots inject such adversarial links into the social graph, the resulting spurious message aggregation corrupts node representations [Schlichtkrull *et al.*, 2018]. The classifier then receives misleading neighborhood information that degrades both predictive accuracy and robustness against adaptive attacks.

To address these challenges, we present **BotVA**, a social **Bot** detection framework that couples **V**ariational feature augmentation with **A**dversarial graph learning. The first module learns class-conditional latent distributions for minority instances and synthesizes semantically coherent features via latent sampling and reconstruction, thereby enlarging minority support in the representation space and reducing decision boundary shifts induced by class imbalance. The adversarial graph module complements this approach by modeling camouflage as structured perturbations on the graph. Specifically, a generator creates deceptive connections between bots and humans to simulate adversarial camouflage strategies, while a discriminator instantiated as a graph transformer with semantic attention learns to identify and downweight such edges by adaptively emphasizing reliable neighborhoods and suppressing suspicious links. Taken together, BotVA provides a unified treatment of sample scarcity and structural noise within a single learning pipeline and can be instantiated with standard GNN backbones. Our key contributions are summarized as follows:

- We formulate bot detection under two practical constraints (class imbalance and camouflaged edges) and introduce a unified framework that addresses both challenges jointly.
- We propose a variational feature augmentation module that learns class-conditional latent distributions of minority instances and synthesizes informative features via latent sampling and reconstruction, effectively expanding minority support and mitigating imbalance-induced boundary shifts.
- We introduce an adversarial graph learning mechanism where a generator simulates deceptive edge formations and a discriminator with graph transformers and semantic attention learns robust representations by adaptively selecting reliable neighborhoods under structural perturbations.
- Extensive experiments on multiple real-world datasets demonstrate state-of-the-art performance, strong robustness under varying imbalance ratios and edge perturbations, and competitive results with limited supervision.

2 Related Work

2.1 Feature-based Methods

Early social bot detection relied on feature engineering from user metadata [Alothali *et al.*, 2021] or text content [Hayawi *et al.*, 2022], combined with traditional classifiers [Varol *et al.*, 2018]. Deep learning approaches using CNNs [Fazil *et al.*, 2021] and Bi-LSTM [Alkahtani and Aldhyani, 2021] have since improved detection capability. However, these methods assume balanced class distributions, overlooking that bots are scarce in real networks. To address this imbalance, recent feature-centric graph methods such as OS-GNN [Shi *et al.*, 2024] and HOVER [Ashmore and Chen, 2023] leverage linear interpolation to reduce class bias. However, these approaches build upon conventional resampling techniques like SMOTE [Chawla *et al.*, 2002] or GraphSMOTE [Zhao *et al.*, 2021], which often introduce noise and overfitting. To address these limitations, we propose a variational inference-based augmentation framework that captures minority class latent distributions and synthesizes diverse features to improve decision boundary quality.

2.2 Graph-based Methods

Graph-based approaches detect camouflaged bots by modeling user interactions as graphs [Shi *et al.*, 2025]. Existing methods utilize homogeneous GNNs (e.g., GCN [Kipf and Welling, 2017], GAT [Veličković *et al.*, 2018]) or heterogeneous GNNs (e.g., RGCN [Schlichtkrull *et al.*, 2018]) for feature extraction. Recent advances such as BotDGT [He *et al.*, 2024] incorporate dynamic graphs with temporal modeling to capture evolutionary patterns. However, bots disrupt graph topology via spurious relationships [Yang *et al.*, 2020], creating structural perturbations that corrupt node embeddings. While SEBot [Yang *et al.*, 2024] and BotHP [He *et al.*, 2025] address camouflage through contrastive and heterogeneity-aware learning, these passive defense methods remain inadequate against dynamic structural attacks. In contrast, BotVA frames camouflage as an adversarial attack and filters spurious relationships via dynamic attack-defense cycles, improving robustness against structural perturbations.

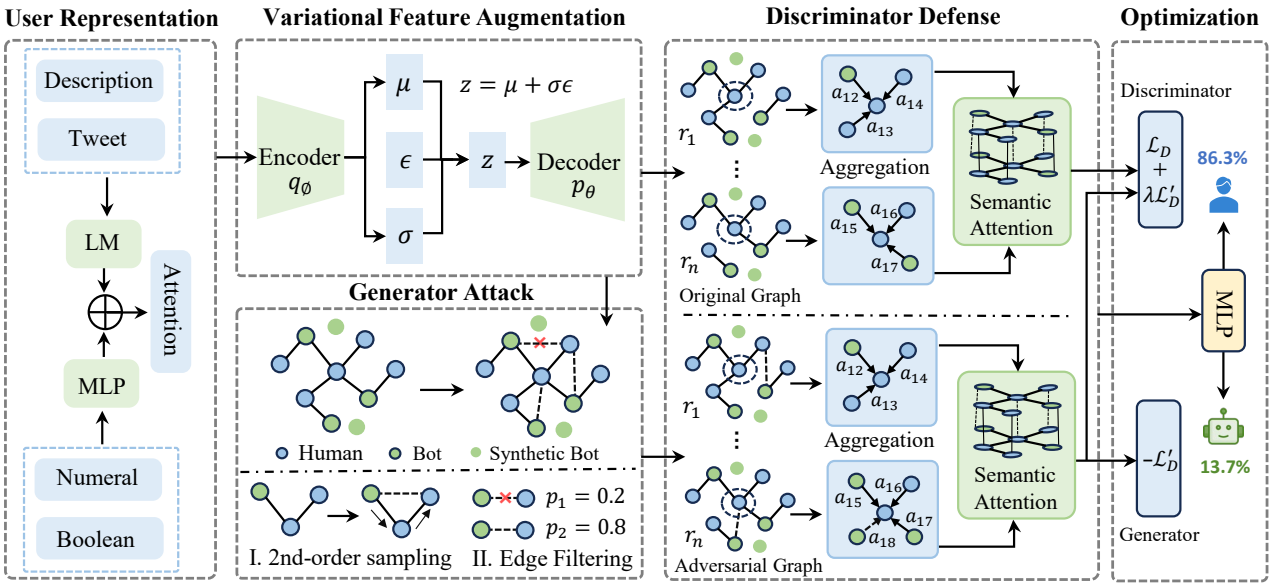

Figure 2: Overview of BotVA. The framework comprises three components: a user representation module, a Variational Feature Augmentation (VFA) module, and an Adversarial Graph Learning (AGL) module that includes a generator and a discriminator.

3 Problem Definition

Social Bot Detection. We formulate social bot detection as semi-supervised binary node classification on a social graph $G = (V, E, X)$, where $V = \{v_i\}_{i=1}^n$ is the set of user nodes with $n = |V|$, $E = \bigcup_{r=1}^R E_r$ is the edge set with R relation types (e.g., follower, friend), and $X \in \mathbb{R}^{n\times\kappa}$ is the node-feature matrix whose i-th row x_i encodes κ-dimensional attributes of v_i (e.g., metadata, textual embeddings). Let $Y = \{y_i\}_{i=1}^n$ be the label set with $y_i \in \{0, 1\}$, where 0 denotes a human and 1 denotes a bot. Given labels on a subset of nodes $V_{\text{train}} \subseteq V$ with Y_{train}, the goal is to learn a node-level predictor $f : (G, Y_{\text{train}}) \to \hat{Y}_{\text{test}}$ that maximizes prediction accuracy for unlabeled nodes $V_{\text{test}} = V \setminus V_{\text{train}}$.

4 Methodology

Figure 2 presents BotVA, which integrates three components: user node representation, Variational Feature Augmentation (VFA), and Adversarial Graph Learning (AGL). The pipeline proceeds as follows. First, the user representation module normalizes raw attributes and encodes heterogeneous signals into a unified node embedding space. Second, the VFA module learns class-conditional latent distributions for minority instances with a variational autoencoder and synthesizes semantically coherent features via latent sampling and reconstruction to alleviate class imbalance. Third, the AGL module models camouflaged edges as structured perturbations on the graph: a generator proposes deceptive connections that emulate bot behavior, and a discriminator, instantiated as a semantic-aware graph transformer, learns to identify and downweight these perturbations by emphasizing reliable neighborhoods. Training alternates between the generator and the discriminator until convergence, after which the discriminator is used for inference on unseen nodes.

4.1 User Node Representation

User information comprises profile attributes and textual content. Profile attributes include numerical and boolean fields, while textual content consists of descriptions and tweets. We encode each source with a dedicated pathway: numerical attributes are z-score normalized and linearly projected to $\mathbf{x}_i^v$; boolean attributes are one-hot encoded and transformed via a linear layer with LeakyReLU to obtain $\mathbf{x}_i^b$; textual content is encoded by a pre-trained language model [Liu *et al.*, 2019] and linearly projected to $\mathbf{x}_i^d$ and $\mathbf{x}_i^t$, respectively.

To fuse multimodal heterogeneous signals and capture high-order interactions among users, we apply a multihead self-attention layer:

$$\begin{aligned} &\mathbf{X} = \{\mathbf{x}_1; \ldots; \mathbf{x}_n \mid \mathbf{x}_i = \text{Concat}(\mathbf{x}_i^v, \mathbf{x}_i^b, \mathbf{x}_i^d, \mathbf{x}_i^t)\}, \\ &\mathbf{Q}_a = \mathbf{X}\mathbf{W}_{a,q},\ \mathbf{K}_a = \mathbf{X}\mathbf{W}_{a,k},\ \mathbf{V}_a = \mathbf{X}\mathbf{W}_{a,v}, \\ &\hat{\mathbf{X}} = \Big\|_{a=1}^{A} \text{Softmax}\left(\frac{\mathbf{Q}_a\mathbf{K}_a^\top}{\sqrt{d_a}}\right)\mathbf{V}_a, \end{aligned} \tag{1}$$

where $d_a = \kappa/A$ is the dimension per head, $(\mathbf{Q}_a, \mathbf{K}_a, \mathbf{V}_a)$ are query, key, and value matrices for the a-th head with learnable parameters $(\mathbf{W}_{a,q}, \mathbf{W}_{a,k}, \mathbf{W}_{a,v})$, and $\|$ concatenates outputs of A heads to yield $\hat{\mathbf{X}}$.

4.2 Variational Feature Augmentation

VFA identifies the specific minority class via training labels and employs a conditional variational autoencoder [Sohn *et al.*, 2015] to model minority class latent distributions and synthesize semantically coherent features, augmenting minority representations without introducing noisy edges. Given a feature $\mathbf{x}_i \in \mathbb{R}^\kappa$ and label $y_i \in \{0, 1\}$, VFA concatenates $\mathbf{x}_i$ with the one-hot encoded label. The encoder estimates the mean $\boldsymbol{\mu}_i$ and log-variance $\log \boldsymbol{\sigma}_i^2$ of the latent variable:

$$\boldsymbol{\mu}_i = f_{\phi,\mu}\left([\mathbf{x}_i \| \mathbf{y}_i]\right),\ \log \boldsymbol{\sigma}_i^2 = f_{\phi,\sigma}\left([\mathbf{x}_i \| \mathbf{y}_i]\right), \tag{2}$$

where $f_{\phi,\mu}(\cdot)$ and $f_{\phi,\sigma}(\cdot)$ are the mean and variance prediction heads. The approximate posterior is:

$$q_\phi\left(\mathbf{z}_i \mid \mathbf{x}_i, \mathbf{y}_i\right) = \mathcal{N}\left(\mathbf{z}_i \mid \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2 \boldsymbol{I}\right), \tag{3}$$

where $\mathbf{z}_i \in \mathbb{R}^\kappa$ is the latent variable. We apply the reparameterization trick [Kingma *et al.*, 2015] for differentiable sampling:

$$\mathbf{z}_i = \boldsymbol{\mu}_i + \boldsymbol{\sigma}_i \boldsymbol{\epsilon},\ \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{4}$$

The decoder reconstructs features $\mathbf{x}_i^*$ by concatenating $\mathbf{z}_i$ with the label embedding:

$$\mathbf{x}_i^* = g_\theta\left([\mathbf{z}_i, \mathbf{y}_i]\right),\ p_\theta\left(\mathbf{x}_i \mid \mathbf{z}_i, \mathbf{y}_i\right) = \mathcal{N}\left(\mathbf{x}_i^*, \sigma_x^2 \mathbf{I}\right), \tag{5}$$

where $g_\theta(\cdot)$ is a multi-layer decoder. Virtual nodes inherit minority labels and are generated iteratively to achieve class balance. The optimization objective is:

$$\begin{aligned}\mathcal{L}_{\mathrm{VFA}}(\phi, \theta) = \mathbb{E}_{q_\phi(\mathbf{z}_i \mid \mathbf{x}_i, \mathbf{y}_i)}\left[-\log p_\theta\left(\mathbf{x}_i \mid \mathbf{z}_i, \mathbf{y}_i\right)\right] \\ + \beta D_{\mathrm{KL}}(q_\phi\left(\mathbf{z}_i \mid \mathbf{x}_i, \mathbf{y}_i\right) \| p(\mathbf{z}))\ ,\end{aligned} \tag{6}$$

where $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the prior distribution over the latent variables, $D_{\mathrm{KL}}(\cdot\|\cdot)$ denotes KL divergence, and β balances reconstruction fidelity against regularization.

4.3 Adversarial Graph Learning

Camouflage in social networks manifests as structured perturbations of the graph. Malicious bots introduce deceptive connections that corrupt message passing in GNNs and mislead downstream classifiers. Static filtering or heuristic edge reweighting offers limited protection against evolving camouflage patterns. We therefore introduce AGL that formulates detection under camouflage as a generator-discriminator game on the graph.

Generator Attack. The generator constructs sparse perturbations by connecting bots to nearby human communities. Target endpoints are restricted to second-order human neighbors, excluding VFA-generated nodes, to maintain stealth. Let $V_b \subseteq V$ and $V_h = V \setminus V_b$ denote bot and human node sets. For each bot $v_i \in V_b$, the candidate interaction set is:

$$\mathcal{C}\left(v_i\right) = \left\{v_j \mid v_j \in \mathcal{N}_h^2\left(v_i\right) \text{ and } \left(v_i, v_j\right) \notin E\right\}, \tag{7}$$

where $\mathcal{N}_h^2(v_i)$ is the second-order human neighbor set. A two-layer MLP predicts connection probability p_{ij} for each candidate edge:

$$p_{ij} = \sigma\left(\mathbf{W}_2\left(\mathbf{W}_1\left[\mathbf{x}_i \| \mathbf{x}_j\right] + b_1\right) + b_2\right)\ . \tag{8}$$

Motivated by [Jang *et al.*, 2017], we cap adversarial edges via Top-k selection for sparsity and use Gumbel-Sigmoid to obtain differentiable weights s_{ij} for backpropagation:

$$s_{ij} = \sigma\left(\frac{\log p_{ij} - \log(1 - p_{ij}) + g_{ij}}{\tau}\right), \tag{9}$$

where $g_{ij} \sim \mathrm{Gumbel}(0, 1)$ represents independent noise, τ is the temperature parameter, and $\sigma(\cdot)$ is the sigmoid function. To reflect heterogeneous relation semantics in real-world social networks, perturbations from high-influence bots (top 10% by degree) are instantiated as bidirectional links, while others are unidirectional. The camouflaged edges merge with the original graph to form the adversarial graph G'.

Discriminator Defense. The discriminator module employs a graph transformer [Yun *et al.*, 2019] with relation-specific multi-head attention. For each node pair (v_i, v_j) under relation r and head c, the attention score is:

$$\begin{aligned}\mathbf{q}_{c,i}^r{}^{(l)} &= \mathbf{W}_{c,q}^r{}^{(l)} \cdot \mathbf{x}_i^{(l-1)} + b_{c,q}^r{}^{(l)}, \\ \mathbf{k}_{c,j}^r{}^{(l)} &= \mathbf{W}_{c,k}^r{}^{(l)} \cdot \mathbf{x}_j^{(l-1)} + b_{c,k}^r{}^{(l)}, \\ \alpha_{c,ij}^r{}^{(l)} &= \frac{\left\langle \mathbf{q}_{c,i}^r{}^{(l)}, \mathbf{k}_{c,j}^r{}^{(l)} \right\rangle}{\sum_{j \in \mathcal{N}^r(i)} \left\langle \mathbf{q}_{c,i}^r{}^{(l)}, \mathbf{k}_{c,j}^r{}^{(l)} \right\rangle},\end{aligned} \tag{10}$$

where $\langle \mathbf{q}, \mathbf{k} \rangle = \exp(\mathbf{q}^T \mathbf{k} / \sqrt{d_c})$, $d_c = \kappa / C$, and $\mathcal{N}^r(i)$ is the neighborhood of v_i under relation r. Node representations are aggregated as:

$$\begin{aligned}\mathbf{v}_{c,j}^r{}^{(l)} &= \mathbf{W}_{c,v}^r{}^{(l)} \cdot \mathbf{x}_j^{(l-1)} + b_{c,v}^r{}^{(l)}, \\ \mathbf{h}_i^{r(l)} &= \Big\Vert_{c=1}^{C} \Big(\sum_{j \in \mathcal{N}^r(i)} \Big(\alpha_{c,ij}^r{}^{(l)} \cdot \boldsymbol{v}_{c,j}^r{}^{(l)} \Big) \Big),\end{aligned} \tag{11}$$

where C is the number of attention heads. A semantic attention network integrates cross-relation features:

$$\beta_m^{r(l)} = \mathrm{Softmax}\Big(\frac{1}{|V|} \sum_{i \in V} q_m^{(l)^T} \cdot \tanh(\mathbf{W}_m^{(l)} \cdot \mathbf{h}_i^{r(l)})\Big), \tag{12}$$

where $\beta_m^{r(l)}$ is the weights at semantic head m, $\mathbf{W}_m^{(l)}$ and $q_m^{(l)}$ are learnable parameters. The final node representation is:

$$\mathbf{x}_i^{(l)} = \frac{1}{M} \sum_{m=1}^{M} \Big(\sum_{r \in R} \beta_m^{r(l)} \cdot \mathbf{h}_i^{r(l)} \Big), \tag{13}$$

where M counts semantic attention heads. A linear layer with Softmax yields predicted labels, optimized via cross-entropy:

$$\begin{aligned}\hat{y}_i &= \mathrm{Softmax}\left(\mathbf{W}_x \mathbf{x}_i + \mathbf{b}_x\right), \\ \mathcal{L}_{\mathrm{AGL}} &= -\sum_{i=1}^{n} \left[y_i \log \hat{y}_i + (1 - y_i) \log\left(1 - \hat{y}_i\right)\right],\end{aligned} \tag{14}$$

where $y_i \in \{0, 1\}$ is the ground-truth label.

4.4 Learning and Optimization

We adopt a two-stage training strategy for high-quality minority synthesis and stable adversarial optimization.

Stage 1 (VFA Pretraining). The VFA module is trained by minimizing $\mathcal{L}_{\mathrm{VFA}}$ to synthesize minority nodes without noisy edges, expanding minority representations to mitigate class imbalance. VFA parameters are then frozen to prevent interference with adversarial optimization.

Stage 2 (AGL Alternating Optimization). Training alternates between discriminator and generator. With the generator fixed, adversarial edges form G', and the discriminator minimizes $(\mathcal{L}_{AGL} + \lambda \mathcal{L}'_{AGL})$ on both G and G'. With the discriminator fixed, the generator minimizes $-\mathcal{L}'_{AGL}$ to maximize classification difficulty. After AGL converges, the discriminator is frozen and deployed as the final detector.

This framework ensures VFA generates minority samples before AGL fortifies robustness against camouflaged edges.

Methods	Dataset	Cresci-15		TwiBot-20		MGTAB	
	Metrics	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
Feature-based	Yang *et al.*	78.42±0.46	78.29±0.35	82.37±0.51	86.17±0.48	-	-
	Wei *et al.*	96.34±0.68	81.95±0.71	70.44±0.29	53.68±0.12	-	-
	Varol *et al.*	93.44±0.36	95.16±0.28	78.29±0.43	80.79±0.53	-	-
	SATAR	94.26±0.47	95.19±0.32	84.18±0.59	86.04±0.56	-	-
Graph-based	GCN	96.33±0.41	96.45±0.37	77.93±0.74	80.87±0.63	84.17±1.28	78.69±1.40
	GAT	96.68±0.39	96.28±0.45	81.13±1.28	82.09±1.32	86.48±1.17	82.35±1.56
	RGT	97.82±0.38	98.14±0.31	86.48±0.59	87.68±0.62	90.13±0.74	86.65±0.83
	BotRGCN	96.98±0.55	97.74±0.52	85.14±0.28	85.87±0.34	89.57±1.10	86.36±1.05
	BotMoE	98.73±0.62	<u>98.87±0.28</u>	86.76±0.53	87.85±0.46	89.31±0.64	84.13±0.52
	CAGCL	97.54±0.43	98.58±0.38	86.83±0.55	88.04±0.61	89.12±0.46	85.25±0.59
	BotBR	98.46±0.31	98.73±0.57	86.95±0.41	88.15±0.60	90.38±0.64	86.58±0.53
	BotDGT	98.71±0.59	98.53±0.46	86.87±0.45	87.98±0.58	-	-
	OS-GNN	96.83±0.40	96.48±0.38	84.48±0.43	84.76±0.75	86.74±0.36	86.36±0.63
	HOVER	97.31±0.53	96.94±0.67	85.16±0.60	85.32±0.72	87.21±0.37	<u>86.92±0.34</u>
	SEBot	<u>98.89±0.47</u>	98.71±0.50	86.96±0.34	87.82±0.45	<u>90.86±0.47</u>	85.48±0.36
	BotHP	98.71±0.45	98.64±0.52	<u>87.12±0.57</u>	<u>88.24±0.65</u>	90.53±0.62	85.19±0.54
Ours	BotVA	**99.74±0.23**	**99.62±0.28**	**87.48±0.25**	**88.67±0.36**	**91.74±0.31**	**89.40±0.39**

Table 1: Performance comparison on Cresci-15, TwiBot-20, and MGTAB. Results are averaged over five independent runs and reported as mean ± std%. "–" indicates that the corresponding method is not applicable to MGTAB. The best and second-best results are highlighted in **bold** and <u>underline</u>, respectively.

5 Experiments

5.1 Experimental Settings

Datasets. We evaluate on three widely used social bot detection benchmarks: Cresci-15 [Cresci *et al.*, 2015], TwiBot-20 [Feng *et al.*, 2021b], and MGTAB [Shi *et al.*, 2025]. For Cresci-15 and TwiBot-20, we adopt the user-profile and textual features released by [Feng *et al.*, 2021c]; for MGTAB, we follow the extraction protocol in [Yang *et al.*, 2024].

Baselines. We compare BotVA with representative feature-based and graph-based methods. Feature-based baselines include Yang *et al.* [Yang *et al.*, 2020], Wei *et al.* [Wei and Nguyen, 2019], Varol *et al.* [Varol *et al.*, 2017], and SATAR [Feng *et al.*, 2021a]. Graph-based competitors include GCN [Kipf and Welling, 2017], GAT [Veličković *et al.*, 2018], RGT [Feng *et al.*, 2022], BotRGCN [Feng *et al.*, 2021c], BotMoE [Liu *et al.*, 2023], CAGCL [Wei *et al.*, 2025], BotBR [Lin and Zhou, 2025], BotDGT [He *et al.*, 2024], OS-GNN [Shi *et al.*, 2024], HOVER [Ashmore and Chen, 2023], SEBot [Yang *et al.*, 2024], and BotHP [He *et al.*, 2025].

5.2 Performance Comparison

We evaluate BotVA against representative baselines and summarize the results in Table 1.

Overall. BotVA achieves the best results on all three datasets. The consistent gains across varying dataset scales and topological characteristics confirm that jointly addressing class imbalance and camouflaged edges yields reliable improvements.

Cresci-15. BotVA reaches 99.74% accuracy and 99.62% F1-score, improving upon SEBot and BotMoE by 0.85% in accuracy and 0.75% in F1-score, respectively. As a moderately sized dataset with relatively clean structure, performance here primarily reflects representation quality. The variational feature augmentation effectively expands minority coverage, producing cleaner decision boundaries.

TwiBot-20. BotVA attains 87.48% accuracy and 88.67% F1-score, outperforming the strongest baseline BotHP. Small standard deviations indicate stable training despite larger graph size and heterogeneous relations. The adversarial graph module proves particularly important on this dataset, as the discriminator learns to downweight deceptive edges while preserving informative neighborhood signals.

MGTAB. BotVA achieves 91.74% accuracy and 89.40% F1-score, outperforming the second-best methods by 0.88% and 2.48%, respectively. This dataset exhibits pronounced label skew and substantial edge noise. The combination of variational minority synthesis and adversarial training effectively improves recall on rare bot instances while maintaining high precision.

5.3 Ablation Study

To quantify component contributions, we conduct controlled ablations on TwiBot-20 and MGTAB with fixed data splits and optimization settings. We evaluate five variants: (i) w/o MA removes multi-head attention in the node encoder, (ii) w/o VFA disables the VFA module, (iii) w/o SAN discards the semantic attention network, (iv) w/o GEN omits the generator, and (v) w/o AGL removes the entire AGL module and uses an MLP classifier. Results are reported in Table 2.

Overall Observations. The full BotVA attains the highest accuracy and F1-score on both datasets. Removing any component yields consistent drops, indicating that each module contributes meaningfully.

Multi-Head Attention. Ablating MA produces moderate declines (TwiBot-20 accuracy -0.22%, MGTAB F1-score

-0.21%), showing that parallel attention over modality-specific tokens captures complementary information.

Settings	TwiBot-20		MGTAB	
	Accuracy	F1-score	Accuracy	F1-score
BotVA	**87.48±0.25**	**88.67±0.36**	**91.74±0.31**	**89.40±0.39**
w/o MA	87.26±0.31	88.38±0.23	91.55±0.28	89.19±0.37
w/o VFA	86.86±0.44	88.01±0.46	91.03±0.47	88.68±0.35
w/o SAN	87.03±0.30	88.12±0.34	91.27±0.36	88.94±0.42
w/o GEN	86.64±0.33	87.86±0.21	90.13±0.45	88.46±0.49
w/o AGL	85.65±0.48	85.83±0.35	87.44±0.31	86.19±0.37

Table 2: Ablation study of BotVA on TwiBot-20 and MGTAB.

Variational Feature Augmentation. Without VFA, accuracy decreases by 0.62% on TwiBot-20 and 0.71% on MGTAB, with F1-score drops of 0.66% and 0.72%, confirming that class-conditional synthesis effectively expands minority coverage under label skew.

Semantic Attention Network. Removing SAN reduces accuracy by 0.45%-0.47% with F1-score declines of 0.46%-0.55%, indicating that relation-level weighting outperforms uniform averaging across heterogeneous edge types.

Adversarial Generator. Eliminating GEN causes larger drops (TwiBot-20: accuracy -0.84%, F1-score -0.81%; MGTAB: accuracy -1.61%, F1-score -0.94%), indicating that the generator can force the discriminator to learn robust representations by simulating deceptive edge patterns.

Adversarial Graph Learning. Removing the entire AGL module causes the most severe degradation (TwiBot-20: accuracy -1.83%, F1-score -2.84%; MGTAB: accuracy -4.30%, F1-score -3.21%), confirming AGL as the primary driver of robustness against structural perturbations.

Dataset-Specific Effects. Ablations show larger impacts on MGTAB, particularly for GEN (1.61% vs 0.84%) and AGL (4.30% vs 1.83%). This aligns with MGTAB's more pronounced label skew and noisier connectivity, suggesting that adversarial components provide greater benefits when camouflage is prevalent.

5.4 Sensitivity Analysis

We analyze sensitivity to the VFA coefficient $\beta \in \{0.1, 0.3, 0.5, 0.7, 1.0\}$ and AGL loss weight $\lambda \in \{0.1, 0.3, 0.5, 0.7, 1.0\}$ on TwiBot-20 and MGTAB (Figure 3). Performance with respect to β peaks at $\beta = 0.3$ on both datasets (87.48% accuracy on TwiBot-20, 89.40% F1-score on MGTAB). Small β values provide insufficient regularization, leading to a fragmented latent space, while large values over-emphasize the prior constraint, compromising reconstruction fidelity. For λ, which balances supervision from the original and perturbed graphs, optimal performance occurs in $[0.3, 0.5]$. Values below 0.1 provide insufficient adversarial signal, whereas $\lambda = 1.0$ overemphasizes adversarial objectives at the expense of classification accuracy on clean structures. Across all configurations, performance variations remain below 1.4% for accuracy and 1.6% for F1-score, indicating that BotVA is robust to hyperparameter selection within reasonable ranges.

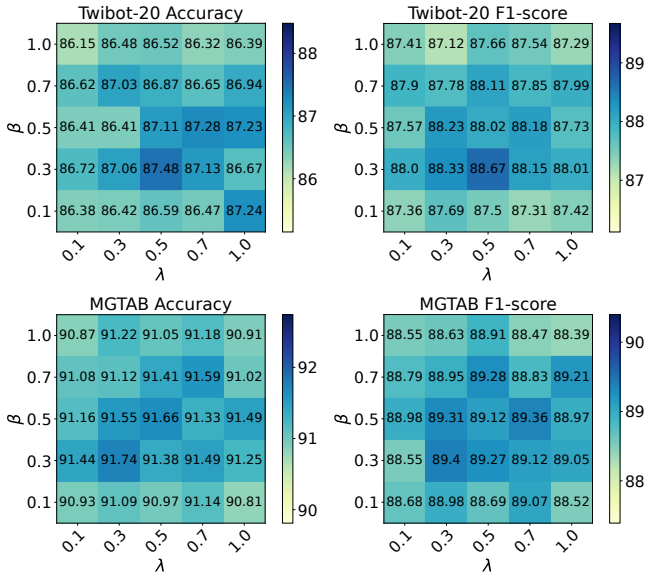

Figure 3: Sensitivity analysis of hyperparameter β and λ.

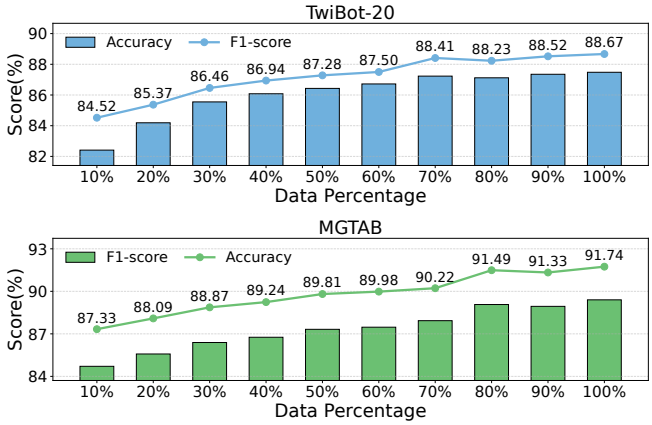

Figure 4: We randomly subsample 10%-100% of the training data, train on each subset, and evaluate on the original test split.

5.5 Data Efficiency Study

To assess performance under limited supervision, we train on random subsets comprising 10%-100% of the original training split while evaluating on unchanged test sets. As shown in Figure 4, BotVA demonstrates strong data efficiency. With only 10% training data, it attains 82%-87% accuracy across datasets, surpassing several baselines trained on full data (Table 1). Performance improves consistently with more labels, saturating earlier on simpler graphs (60%-70%) and later on larger, noisier graphs (80%-90%). This efficiency stems from two complementary mechanisms: VFA expands minority support through class-conditional synthesis, while AGL learns robust representations via adversarial training. Notably, BotVA trained on 70%-90% of data matches or exceeds baselines trained on full datasets, demonstrating practical value in label-scarce scenarios.

5.6 Visualization Analysis

Generated Sample Fidelity. We visualize kernel density-estimate contours for original and VFA-generated minority samples on Cresci-15 and MGTAB (Figure 5). On Cresci-15, the two distributions show substantial overlap with closely aligned density peaks, indicating that VFA captures the underlying minority distribution. On the larger and more heterogeneous MGTAB, distributions are more dispersed, yet generated samples still cover core high-density regions, demonstrating effective minority support expansion under pronounced class imbalance.

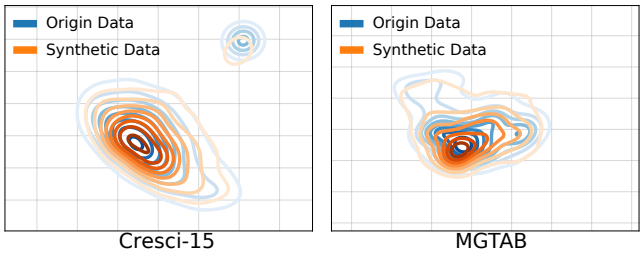

Figure 5: Density-contour visualization of original samples and VFA-generated samples on Cresci-15 and MGTAB datasets.

Robustness to Camouflage Attacks. We evaluate robustness by injecting 0%-25% additional bot-human edges into TwiBot-20. As shown in Figure 6, BotVA maintains the highest F1-score across all perturbation levels. While all methods degrade with increasing attack intensity, degradation rates differ substantially. Methods without heterogeneous relation modeling (e.g., BotMoE) degrade rapidly, while heterogeneity-aware approaches (e.g., BotHP, SEBot) show improved resilience but still lag behind BotVA. These results confirm that simulating camouflage strategies during training yields superior robustness.

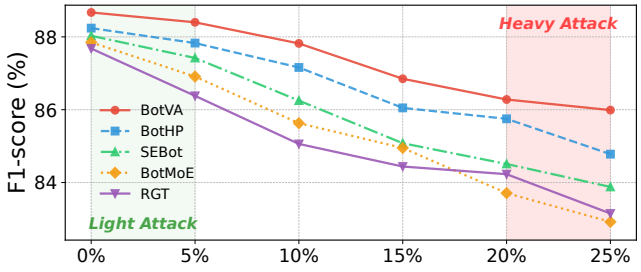

Figure 6: Robustness of BotVA under camouflage attacks on TwiBot-20. We simulate attacks by randomly adding 0%-25% bot-human camouflage edges.

Framework Generality. To verify extensibility, we integrate four representative GNN backbones into BotVA (Figure 7). Results show consistent performance improvements across all architectures. The F1-score gains on MGTAB are particularly pronounced compared to accuracy gains on Cresci-15, indicating that BotVA is especially effective in imbalanced scenarios where VFA enriches minority class representations. This confirms BotVA as a model-agnostic framework capable of empowering standard GNNs.

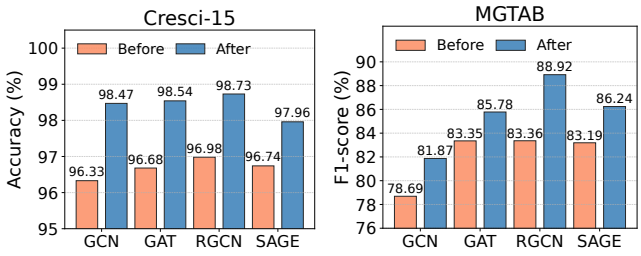

Figure 7: Performance comparison before and after integrating GNN backbones into BotVA (AGL) on Cresci-15 and MGTAB datasets.

Representation Discriminability. We project final-layer embeddings to two dimensions using t-SNE (Figure 8). BotDGT and SEBot display coarse inter-class separation but substantial intra-class overlap. BotHP exhibits improved intra-class cohesion yet maintains weaker inter-class margins. In contrast, BotVA achieves both clear inter-class separation and compact within-class clusters, validating its superiority.

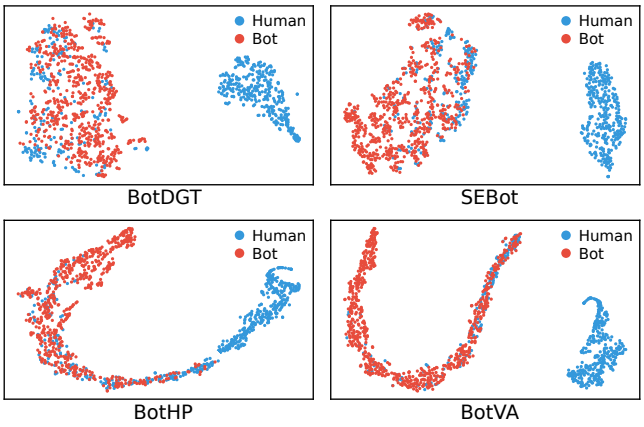

Figure 8: Visualization of account representations on TwiBot-20. Red represents bots and blue represents humans.

6 Conclusion

We presented BotVA, a unified framework for graph-based bot detection that jointly tackles class imbalance and structural noise from camouflaged edges. BotVA uses a conditional variational autoencoder to generate semantically coherent minority-class features, mitigating decision boundary shifts, and introduces adversarial graph learning, where a generator injects deceptive edges while a graph-transformer discriminator with semantic attention learns robust representations. With a two-stage training strategy, BotVA achieves stable sample generation and adversarial optimization. Experiments show that BotVA delivers state-of-the-art performance, strong robustness to structural perturbations, and competitive results under limited supervision, highlighting the value of jointly modeling minority-class distributions and adversarial structural attacks.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62572404 and 62402458), the Guangdong Basic and Applied Basic Research Foundation (2025A1515011565), the China Postdoctoral Science Foundation (2025M784417), the Key Research and Development Program of Shaanxi (2024GX-ZDCYL-01-20), and the Beijing Natural Science Foundation (4262061).

References

[Alarifi *et al.*, 2016] Abdulrahman Alarifi, Mansour Alsaleh, and AbdulMalik Al-Salman. Twitter turing test: Identifying social machines. *Information Sciences*, 372:332–346, 2016.

[Alkahtani and Aldhyani, 2021] Hasan Alkahtani and Theyazn HH Aldhyani. Botnet attack detection by using cnn-lstm model for internet of things applications. *Security and Communication Networks*, 2021(1):3806459, 2021.

[Alothali *et al.*, 2021] Eiman Alothali, Kadhim Hayawi, and Hany Alashwal. Hybrid feature selection approach to identify optimal features of profile metadata to detect social bots in twitter. *Social Network Analysis and Mining*, 11(1):84, 2021.

[Ashmore and Chen, 2023] Bradley Ashmore and Lingwei Chen. HOVER: Homophilic oversampling via edge removal for class-imbalanced bot detection on graphs. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, page 3728–3732, 2023.

[Beskow and Carley, 2018] David M Beskow and Kathleen M Carley. Bot conversations are different: leveraging network metrics for bot detection in twitter. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 825–832, 2018.

[Caldarelli *et al.*, 2020] Guido Caldarelli, Rocco De Nicola, Fabio Del Vigna, Marinella Petrocchi, and Fabio Saracco. The role of bot squads in the political propaganda on twitter. *Communications Physics*, 3(1):81, 2020.

[Chawla *et al.*, 2002] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

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

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

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

[Fazil *et al.*, 2021] Mohd Fazil, Amit Kumar Sah, and Muhammad Abulaish. Deepsbd: a deep neural network model with attention mechanism for socialbot detection. *IEEE Transactions on Information Forensics and Security*, 16:4211–4223, 2021.

[Feng *et al.*, 2021a] Shangbin Feng, Herun Wan, Ningnan Wang, Jundong Li, and Minnan Luo. SATAR: A self-supervised approach to twitter account representation learning and its application in bot detection. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3808–3817, 2021.

[Feng *et al.*, 2021b] Shangbin Feng, Herun Wan, Ningnan Wang, Jundong Li, and Minnan Luo. TwiBot-20: A comprehensive twitter bot detection benchmark. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 4485–4494, 2021.

[Feng *et al.*, 2021c] Shangbin Feng, Herun Wan, Ningnan Wang, and Minnan 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*, pages 236–239, 2021.

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

[Hayawi *et al.*, 2022] Kadhim Hayawi, Sujith Mathew, Neethu Venugopal, Mohammad M Masud, and Pin-Han Ho. DeeProBot: a hybrid deep neural network model for social bot detection based on user profile data. *Social Network Analysis and Mining*, 12(1):43, 2022.

[He *et al.*, 2024] Buyun He, Yingguang Yang, Qi Wu, Hao Liu, Renyu Yang, Hao Peng, Xiang Wang, Yong Liao, and Pengyuan Zhou. Dynamicity-aware social bot detection with dynamic graph transformers. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 2024.

[He *et al.*, 2025] Buyun He, Xiaorui Jiang, Qi Wu, Hao Liu, Yingguang Yang, and Yong Liao. Boosting bot detection via heterophily-aware representation learning and prototype-guided cluster discovery. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 860–871, 2025.

[Jang *et al.*, 2017] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

[Javed *et al.*, 2024] Danish Javed, NZ Jhanjhi, Navid Ali Khan, Sayan Kumar Ray, Alanoud Al Mazroa, Farzeen Ashfaq, and Shampa Rani Das. Towards the future of bot detection: A comprehensive taxonomical review and challenges on twitter/x. *Computer Networks*, 254:110808, 2024.

[Johnson and Khoshgoftaar, 2019] Justin M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):1–54, 2019.

[Kingma *et al.*, 2015] Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. *Advances in Neural Information Processing Systems*, 28, 2015.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

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

[Lin and Zhou, 2025] Qilong Lin and Jingya Zhou. BotBR: Social bot detection with balanced feature fusion and reliability-enhanced graph learning. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 392–402, 2025.

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

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

[Schlichtkrull *et al.*, 2018] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607, 2018.

[Schnebly and Sengupta, 2019] James Schnebly and Shamik Sengupta. Random forest twitter bot classifier. In *2019 IEEE 9th Annual Computing and Communication Workshop and Conference*, pages 0506–0512, 2019.

[Shao *et al.*, 2018] Chengcheng Shao, Giovanni Luca Ciampaglia, Onur Varol, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. The spread of low-credibility content by social bots. *Nature Communications*, 9(1):4787, 2018.

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

[Shi *et al.*, 2025] Shuhao Shi, Kai Qiao, Zihao Liu, Jie Yang, Chen Chen, Jian Chen, and Bin Yan. MGTAB: A multi-relational graph-based twitter account detection benchmark. *Neurocomputing*, 647:130490, 2025.

[Sohn *et al.*, 2015] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in Neural Information Processing Systems*, 28, 2015.

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

[Varol *et al.*, 2017] Onur Varol, Emilio Ferrara, Clayton Davis, Filippo Menczer, and Alessandro Flammini. Online human-bot interactions: Detection, estimation, and characterization. In *Proceedings of the International AAAI Conference on Web and Social Media*, pages 280–289, 2017.

[Varol *et al.*, 2018] Onur Varol, Clayton A Davis, Filippo Menczer, and Alessandro Flammini. Feature engineering for social bot detection. In *Feature Engineering for Machine Learning and Data Analytics*, pages 311–334. 2018.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[Wei and Nguyen, 2019] Feng Wei and Uyen Trang 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*, pages 101–109, 2019.

[Wei *et al.*, 2025] Kaihang Wei, Min Teng, Haotong Du, Songxin Wang, Jinhe Zhao, and Chao Gao. CAGCL: A community-aware graph contrastive learning model for social bot detection. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 3282–3291, 2025.

[Yang *et al.*, 2020] Kai-Cheng Yang, Onur Varol, Pik-Mai Hui, and Filippo Menczer. Scalable and generalizable social bot detection through data selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1096–1103, 2020.

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

[Yun *et al.*, 2019] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[Zhao *et al.*, 2021] Tianxiang Zhao, Xiang Zhang, and Suhang Wang. GraphSMOTE: Imbalanced node classification on graphs with graph neural networks. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 833–841, 2021.