Dynamicity-aware Social Bot Detection with Dynamic Graph Transformers

Buyun He[1,*], **Yingguang Yang**[1,*], **Qi Wu**[1], **Hao Liu**[1], **Renyu Yang**[2]
Hao Peng[2,3,†], **Xiang Wang**[1], **Yong Liao**[1,†], **Pengyuan Zhou**[4]
[1]University of Science and Technology of China
[2]Beihang University
[3]Harbin Engineering University
[4]Aarhus university
{byhe, dao, qiwu4512, rcdchao}@mail.ustc.edu.cn, {renyu.yang, penghao}@buaa.edu.cn,
xiangwang1223@gmail.com, yliao@ustc.edu.cn, pengyuan.zhou@ece.au.dk

Abstract

Detecting social bots has evolved into a pivotal yet intricate task, aimed at combating the dissemination of misinformation and preserving the authenticity of online interactions. While earlier graph-based approaches, which leverage topological structure of social networks, yielded notable outcomes, they overlooked the inherent dynamicity of social networks – In reality, they largely depicted the social network as a static graph and solely relied on its most recent state. Due to the absence of dynamicity modeling, such approaches are vulnerable to evasion, particularly when advanced social bots interact with other users to camouflage identities and escape detection. To tackle these challenges, we propose BotDGT, a novel framework that not only considers the topological structure, but also effectively incorporates dynamic nature of social network. Specifically, we characterize a social network as a dynamic graph. A structural module is employed to acquire topological information from each historical snapshot. Additionally, a temporal module is proposed to integrate historical context and model the evolving behavior patterns exhibited by social bots and legitimate users. Experimental results demonstrate the superiority of BotDGT against the leading methods that neglected the dynamic nature of social networks in terms of accuracy, recall, and F1-score.

1 Introduction

As social networks become integrated into people's daily routines, there is a prevalent occurrence of program-controlled bots masquerading as legitimate users for malicious purposes [Subrahmanian *et al.*, 2016]. Social bots engage in detrimental activities such as propagating misinformation [Varol *et al.*, 2017; Gao *et al.*, 2023], manipulating public opinion [Cui *et al.*, 2020], interfering in elections [Rossi *et al.*, 2020] and promoting extremist ideologies [Ferrara *et al.*, 2016]. It is therefore imperative to effectively detect social bots to mitigate the detrimental societal and economic impact and to preserve the integrity of social network information.

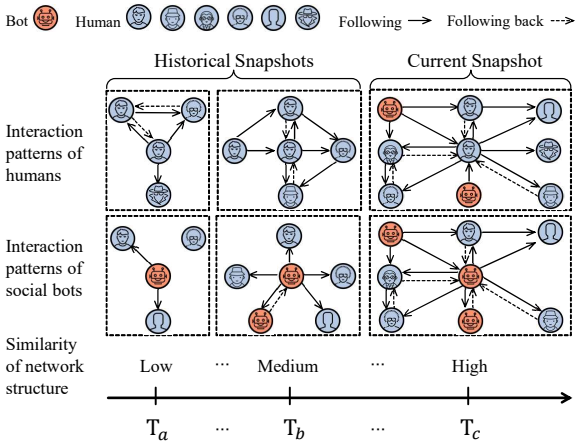

Figure 1: Dynamic nature of social network

Traditional techniques for bot detection are largely based on features, requiring extraction of either numerical feature from user information [Yang *et al.*, 2013] or semantic features from textual information [Wei and Nguyen, 2019; Dukić *et al.*, 2020]. However, bot operators can often bypass bot detection through advanced countermeasures, which is commonly referred to as *bot evolution* [Cresci, 2020]. In fact, the detectability of the feature-based methods is vulnerable to imitation and evasion, as bot operators can effortlessly steal user information from legitimate users or intersperse a few malicious messages with many neutral ones [Feng *et al.*, 2022b]. As a result, such methods are inaccurate in spotting disguised social bots. With the advancements in graph neural networks, some researchers employed graph-based methods [Wu *et al.*, 2023; Feng *et al.*, 2022a; Yang *et al.*, 2023a] to identify the disguised social bots. They typically assume that the network structure of social bots generally differs from that of legitimate users. For instance, social bots tend to have sparser connections and randomly select users to interact with, whereas human beings prefer to con-

*The authors contributed equally to this work.
†Corresponding authors.

nect with others who share similar characteristics [Yang *et al.*, 2013]. These graph-based methods are among top performers by leveraging the topological structure of social networks for bot detection. However, most of the existing graph-based detection methods interpret the social network as a *static* graph and fail to acquire the dynamic nature of social networks. As shown in Figure 1, there still remain two intractable issues:

Deficiency in utilizing historical interaction graph context. Similar to the case of evading detection from feature-based methods by forging numerical or semantic features, the ever-evolving social bots are meticulously engineered to interact with legitimate users and mimic their network structures [Cresci, 2020] to escape graph-based detection. However, despite the structure of social network has changed, the discrepancies in the previous interaction graph between social bots and benign users could reveal the deception of social bots and uncover their true identity. Unfortunately, conventional approaches upon *static* graphs solely rely on the last state of the social network and overlook the valuable historical interaction graph context. Consequently, if the social bots have *already* completed their disguise, it is challenging for static graph based methods to distinguish benign users from the evolved social bots.

Limitation of modeling evolving behavior patterns. Social bots evolve over time, evading detection by dynamically adapting their actions, strategies, or interaction patterns to mimic legitimate users. In contrast, genuine users do not require such adaptations and exhibit different evolution of behavior patterns compared to social bots. Discovering the evolving behavior patterns may enhance the effectiveness of social network modeling [Liu *et al.*, 2020]. Nevertheless, static graph based methods fall short of modeling the distinct evolving behavior patterns of social bots and legitimate users, leading to erroneous results when conducting bot detection tasks.

To overcome the limitations above, we propose a new framework called BotDGT (**Bot** detection with **D**ynamic **G**raph **T**ransformers). The key insight is to introduce the dynamicity modeling of social network for bot detection. To this end, BotDGT depicts a social network as a *dynamic* graph for modeling historical interaction graph contexts and discerning the evolving behavior patterns. Specifically, we interpret users and interactions as nodes and edges, respectively, to generate a batch of snapshots at a fixed time interval for a given social network. A structural module that employs message-passing mechanism is proposed to model the topological structure of each historical snapshot. Additionally, a temporal module based on self-attention mechanism is further employed to incorporate historical contexts and exploit the distinct behavior patterns evolution exhibited by social bots and legitimate users. Overall, our contributions are summarized as follows:

- To the best of our knowledge, we are the first to characterize a social network as a dynamic graph and effectively identify the ever-evolving social bots that disguise themselves through adapting their behavior patterns.
- We introduce a novel bot detection framework to consider both topological structure and the dynamic nature of social networks to enhance the performance of bot detection.
- We conduct comprehensive experiments on two benchmarks for bot detection, which demonstrates the superior performance of BotDGT compared to the leading methods in terms of accuracy, recall and F1-score. Further experiments substantiate the effectiveness of incorporating the dynamic nature of social networks for bot detection.

2 Preliminaries

2.1 Related Work

Social Bot Detection

Early methods for social bot detection are predominantly feature-based. Researchers extracted numerical features from user information and fed them into machine learning models for classification [Lee *et al.*, 2011; Mazza *et al.*, 2019; Yang *et al.*, 2020] or anomaly detection [Miller *et al.*, 2014]. Some studies employed natural language processing techniques to encode textual information, capturing semantic features to enhance the feature-based methods [Hayawi *et al.*, 2022; Dukić *et al.*, 2020; Yang *et al.*, 2023b]. However, newer generations of social bots may forge numerical features or semantic features, either by stealing legitimate users' information or interspersing malicious messages among benign ones, to evade feature-based detection.

With the advancements in graph neural networks, some graph-based methods leveraged the topological structure of social networks for bot detection [Pham *et al.*, 2022; Shi *et al.*, 2023; Peng *et al.*, 2024; Zeng *et al.*, 2024]. The study [Ali Alhosseini *et al.*, 2019] takes the first attempt to introduce graph convolutional networks to aggregate user information from neighboring nodes for bot detection. Subsequent investigations modeled the heterogeneity of social networks and yielded leading performance [Feng *et al.*, 2021b; Feng *et al.*, 2022a]. However, these methods interpreted the social networks as static graphs and neglected the intrinsic dynamicity of real-world social networks, thereby falling short of detecting evolving social bots that adapt strategies to mimic legitimate users' network structure [Cresci, 2020]. To this end, we build upon previous research and present a dynamicity-aware bot detection framework. It incorporates historical context and exploits the evolution of user behavioral patterns, aiming at enhancing the performance of bot detection.

Dynamic Graph Neural Network

Dynamic graphs capture temporal information through time-based dimensions [Skarding *et al.*, 2021; Peng *et al.*, 2021; Sun *et al.*, 2021]. Previous research in graph representation learning has predominantly concentrated on static scenarios, presuming fixed topological structures. However, real-world graphs, including social networks [Alvarez-Rodriguez *et al.*, 2021; Wang *et al.*, 2021], exhibit continual evolution and dynamic characteristics over time. Dynamic graph neural networks are designed to capture this dynamic nature and are widely adopted in various tasks, including link prediction [Xie *et al.*, 2021; Chen *et al.*, 2022; Sankar *et al.*, 2020; Zhang *et al.*, 2023], anomaly detection [Cai *et al.*, 2021; Guo *et al.*, 2022], and node classification [Kim *et al.*, 2022; Pareja

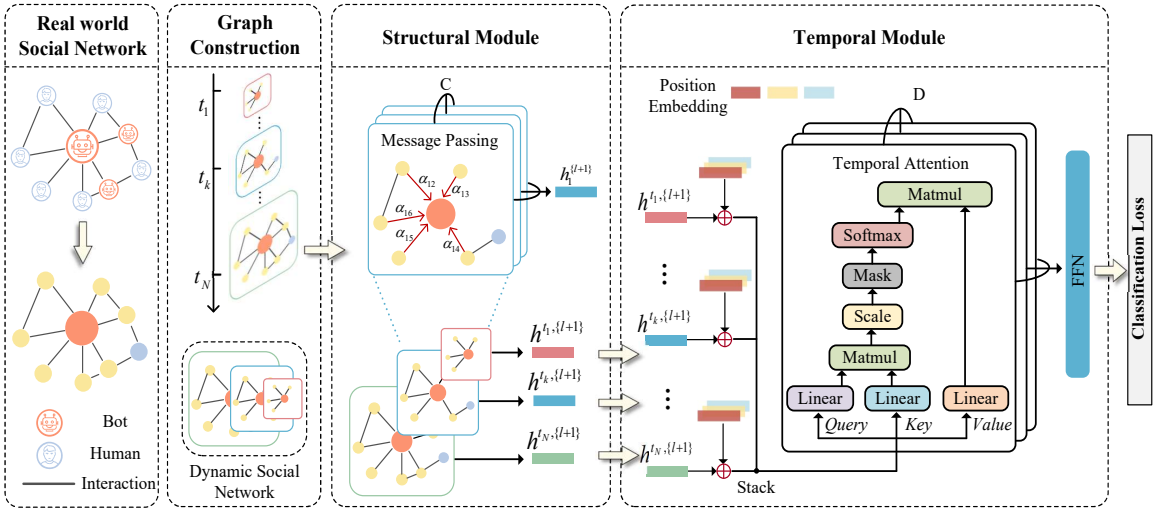

Figure 2: Overview of our proposed BotDGT framework.

et al., 2020; Xu *et al.*, 2020]. Drawing upon the previous works that employed recurrent neural networks [Chen *et al.*, 2022; Zhou *et al.*, 2020] and attention mechanisms [Sankar *et al.*, 2020; Xu *et al.*, 2020] to model dynamic graphs, we propose a novel approach that utilizes a self-attention mechanism to leverage the dynamic nature of social network, thereby enhancing the effectiveness of bot detection.

2.2 Problem Definition

In this paper, we depict the social network as a dynamic graph that changes over time and capture the dynamic nature of the social network to improve the performance of the bot detection model. In this part, we first define the dynamic social network and then formulate the problem.

Definition (Dynamic Social Network)**.** *A dynamic social network is depicted as a graph* $G = \{G^{t_1}, G^{t_2}, ..., G^{t_N}\}$ *with a series of network snapshots over time.* $G^{t_k} = (V^{t_k}, E^{t_k})$ *represents the snapshot of a given social network graph at the timestamp* t_k*, where* V^{t_k}*,* E^{t_k} *are users and interactions respectively observed at timestamp* t_k*.*

To align with the previous studies, we treat bot detection as a binary classification problem, i.e., users are classified into human (y = 0) or bot (y = 1). We formulate the problem of bot detection in dynamic social networks as follows:

Problem (Bot Detection in Dynamic Social Network)**.** *Given a dynamic social network* $G = \{G^{t_1}, G^{t_2}, ..., G^{t_N}\}$*, the problem is to find an encoding function* $f : v \rightarrow \hat{y}$ *for each node* $v \in V^{t_k}$ *at the timestamp* $t_k \in \{t_1, t_2, ..., t_N\}$*, such that* $\hat{y}$ *approximates the ground truth* y *to maximize prediction accuracy.*

3 Methodology

As shown in Figure 2, BotDGT comprises two modules to primarily capture the topological structure and dynamicity of social networks. Specifically, we produce snapshots of the social network at a certain time interval. For each snapshot, the structural module aggregates the features of neighboring nodes and generates temporary node representations containing the snapshot's topology information. Additionally, the temporal module integrates historical context and exploits changes in behavior patterns over time. The resultant node representations are subsequently utilized to differentiate social bots from genuine users.

3.1 Constructing Dynamic Graph

We first construct a dynamic social network $G = \{G^{t_1}, G^{t_2}, ..., G^{t_N}\}$ at a specific time interval Δ_t. We then acquire the node representation using the information encoding procedure established in the state-of-the-art techniques from recent study [Feng *et al.*, 2022a]. A fully-connected layer is used to transform $\boldsymbol{x}_i^{t_k}$, the feature of user i at timestamp t_k, into the initial user vector $\boldsymbol{h}_i^{t_k,\{0\}}$:

$$\boldsymbol{h}_i^{t_k,\{0\}} = \sigma(\boldsymbol{W}_I \boldsymbol{x}_i^{t_k} + \boldsymbol{b}_I), \tag{1}$$

where $\boldsymbol{W}_I$, $\boldsymbol{b}_I$ are trainable parameters and $\sigma(\cdot)$ is a nonlinear activation function.

3.2 Modeling Topological Structure

Upon generating a dynamic social network and initial user vectors, we propose a structural module that leverages a message-passing mechanism [Gilmer *et al.*, 2017] to effectively model the topological structure of each snapshot. This structural module takes a snapshot $G^{t_k} = (V^{t_k}, E^{t_k})$ and a set of initial user vector $\{\boldsymbol{h}_i^{t_k}, \forall v_i \in V^{t_k}\}$ at timestamp t_k as input. The output includes a new set of temporary node representations $\{\boldsymbol{s}_i^{t_k}, \forall v_i \in V^{t_k}\}$, which captures the structural information of the snapshot at t_k.

Graph attention networks [Veličković *et al.*, 2017] have shown superior performance when tackling graph data, by specifying different weights to different nodes within a neighborhood. Inspired by the Transformer architecture [Vaswani *et al.*, 2017], we adopt a scaled dot-product attention mechanism for provisioning each node with the ability to learn the importance of its neighbors within a particular snapshot.

Specifically, for a given a node pair (v_i, v_j) in snapshot G^{t_k}, the attention weight can be calculated as such:

$$\begin{aligned} \boldsymbol{q}_i^{t_k,\{l\}} &= \boldsymbol{W}_q^{\{l\}} \cdot \boldsymbol{h}_i^{t_k,\{l\}} + \boldsymbol{b}_q^{\{l\}}, \\ \boldsymbol{k}_j^{t_k,\{l\}} &= \boldsymbol{W}_k^{\{l\}} \cdot \boldsymbol{h}_j^{t_k,\{l\}} + \boldsymbol{b}_k^{\{l\}}, \\ \alpha_{ij}^{t_k,\{l\}} &= \frac{\left\langle \boldsymbol{q}_i^{t_k,\{l\}}, \boldsymbol{k}_j^{t_k,\{l\}} \right\rangle}{\sum_{u \in \mathcal{N}_{(i)}^{t_k}} \left\langle \boldsymbol{q}_i^{t_k,\{l\}}, \boldsymbol{k}_u^{t_k,\{l\}} \right\rangle}, \end{aligned} \tag{2}$$

where $\{l\}$ denotes the l-th layer of structural module and $\mathcal{N}_{(i)}^{t_k}$ denotes the neighborhood of node v_i at the timestamp t_k. The initial user vector $\boldsymbol{h}_i^{t_k,\{l\}}$ and $\boldsymbol{h}_j^{t_k,\{l\}}$ are transformed into a query vector $\boldsymbol{q}_i^{t_k,\{l\}}$ and a key vector $\boldsymbol{k}_j^{t_k,\{l\}}$. The attention weight $\alpha_{ij}^{t_k,\{l\}}$, which indicates the contribution of node v_j to node v_i at the snapshot G^{t_k}, is calculated by the exponential scale dot product function $\langle q, k \rangle = \exp\left(\frac{\boldsymbol{q}\boldsymbol{k}^T}{\sqrt{d}}\right)$, where d is the input embedding dimension.

After obtaining the attention weight, we transform the initial user vector into a value vector and aggregate information from the neighboring nodes of v_i. A multi-head attention mechanism is employed to capture diverse patterns and dependencies in the topological structure of the social network:

$$\begin{aligned} \boldsymbol{v}_j^{t_k,\{l\}} &= \boldsymbol{W}_v^{\{l\}} \cdot \boldsymbol{h}_j^{t_k,\{l\}} + \boldsymbol{b}_v^{\{l\}}, \\ \boldsymbol{h}_i^{t_k,\{l+1\}} &= \Big\Vert_{c=1}^{C} \sigma\left(\sum_{j \in \mathcal{N}_{(i)}^{t_k}} \left(\alpha_{c,ij}^{t_k,\{l\}} . \boldsymbol{v}_{c,j}^{t_k,\{l\}}\right)\right), \end{aligned} \tag{3}$$

where $\Vert$ represents the concatenation operation, $\alpha_{c,ij}^{t_k,\{l\}}$ denotes the attention weight computed by the c-th attention head, and $\boldsymbol{v}_{c,j}^{t_k,\{l\}}$ denotes the corresponding value vector.

It is worth noting that we stack L layers to allow nodes to capture more distant and global dependencies in the topological structure. The output of the last layer in the structural module is denoted as $\boldsymbol{s}$.

3.3 Acquiring Temporal Dynamicity

While BotDGT's structural module can capture topological information from static snapshots, it insufficiently leverages historical context and fails to discover evolving behavior patterns of social bots. Inspired by [Sankar *et al.*, 2020; Ying *et al.*, 2021], we devise a self-attention based temporal module to further make use of the temporal characteristics of social networks for bot detection. The temporal module takes as inputs a sequence of temporary representations of node v_i at each timestamp, denoted as $\left\{\boldsymbol{s}_i^{t_1}, \boldsymbol{s}_i^{t_2}, ..., \boldsymbol{s}_i^{t_N}\right\}$. The module outputs a new sequence of user representations $\left\{\hat{\boldsymbol{z}}_i^{t_1}, \hat{\boldsymbol{z}}_i^{t_2}, ..., \hat{\boldsymbol{z}}_i^{t_N}\right\}$, where $\hat{\boldsymbol{z}}_i^{t_k}$ denotes the final representation that contains both topological and temporal feature of node v_i at t_k.

Position Embedding Layer

Since the self-attention mechanism is unaware of the nodes' ordering information, we introduce a position embedding layer to accommodate temporal information in the sequence that can effectively reflect the dynamic nature of social networks. We consider two categories of position embedding – absolute temporal position embedding and evolving temporal position embedding.

First, we embed the absolute temporal position [Gehring *et al.*, 2017] of each snapshot as a basis to capture ordering information as follows:

$$\boldsymbol{p}^{t_k,AT} = E_{AT}(t_k), \tag{4}$$

where $\boldsymbol{p}^{t_k,AT}$ denotes the **A**bsolute **T**emporal position embedding for the timestamp t_k and E_{AT} denotes the trainable absolute temporal position embedding parameter. Note that the absolute temporal position embedding only relies on the order of the snapshot, indicating that the nodes in the same snapshot have the same absolute temporal position embedding, i.e., the absolute temporal position embedding is independent of the nodes' features.

Second, we embed two crucial temporal signals: the local clustering coefficient and bidirectional links ratio. These signals have demonstrated their utility in countering the disguised social bots [Yang *et al.*, 2013] and could reveal the evolving behavior patterns over time.

• *Local Clustering Coefficient (LCC)*: it measures the degree to which a node's neighbors are interconnected. Genuine users typically engage with acquaintances (e.g., friends, family members, and colleagues) who have similar connections and thus form closely-knit communities. By contrast, social bots are usually associated with randomly selected neighbors who lack close connectivity, which results in reduced clustering coefficients when compared with legitimate users. The position embedding of the local clustering coefficient is calculated as follows:

$$LCC(v_i^{t_k}) = \frac{2 * |e_{v_i}^{t_k}|}{k_{v_i}^{t_k} * (k_{v_i}^{t_k} - 1)}, \quad p_i^{t_k,LCC} = E_{LCC}(LCC(v_i^{t_k})), \tag{5}$$

where $|e_{v_i}^{t_k}|$ is the number of edges between neighbors of node v_i at the timestamp t_k, $k_{v_i}^{t_k}$ is the sum of the indegree and outdegree of node v_i at t_k.

• *Bidirectional Links Ratio (BLR)*: it is a metric in social network analysis to assess the reciprocity between an account and its followings [Yang *et al.*, 2013]. A bidirectional link appears when two accounts mutually follow each other. This metric proves particularly useful in distinguishing between genuine users, who often own higher bidirectional link counts due to reciprocal following acquaintances with mutual follow-backs, and social bots, who exhibit lower bidirectional link counts due to their indiscriminate following behavior and lack of reciprocal connections. The position embedding of the bidirectional links ratio is calculated as follows:

$$BLR(v_i^{t_k}) = \frac{N_{blinks}(v_i^{t_k})}{N_{fing}(v_i^{t_k})}, \quad p_i^{t_k,BLR} = E_{BLR}(BLR(v_i^{t_k})), \tag{6}$$

where $N_{blinks}(v_i^{t_k})$ and $N_{fing}(v_i^{t_k})$ denote the numbers of bidirectional links and following interactions.

In summary, the integration of these two categories of position embeddings enables the temporal module to capture essential temporal insights from ordering information and the evolving behavior patterns of social bots.

Temporal Attention Layer

The temporal attention layer starts from gathering the outputs of the structural module and the position embedding layer:

$$\hat{\boldsymbol{s}}_i^{t_k} = \boldsymbol{s}_i^{t_k} + \boldsymbol{p}_i^{t_k,AT} + \boldsymbol{p}_i^{t_k,LCC} + \boldsymbol{p}_i^{t_k,BLR}. \tag{7}$$

Then we pack the representations of node v_i together across the timestamps, which is denoted as $\hat{\boldsymbol{S}}_i \in \mathbb{R}^{T\times F}$. Finally we perform multi-head temporal attention as follows:

$$\begin{gathered}\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i = \hat{\boldsymbol{S}}_i\boldsymbol{W}_q, \hat{\boldsymbol{S}}_i\boldsymbol{W}_k, \hat{\boldsymbol{S}}_i\boldsymbol{W}_v, \\ \hat{\boldsymbol{Z}}_i = \Big\|_{d=1}^{D} softmax(\frac{\boldsymbol{Q}_{d,i}\boldsymbol{K}_{d,i}^T}{\sqrt{F}} + \boldsymbol{Mask}) \cdot \boldsymbol{V}_{d,i},\end{gathered} \tag{8}$$

where $\|$ represents the concatenation operation, $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ are the queries, keys, and values transformed by trainable parameters $\boldsymbol{W}_* \in \mathbb{R}^{F\times F}$ respectively. $\boldsymbol{Mask} \in \mathbb{R}^{T\times T}$ is a sequence mask matrix that makes sure the node at timestamp t_k only attends over its historical node representation. The $\boldsymbol{Mask}$ is defined as follows:

$$Mask_{ab} = \begin{cases} 0 & \text{if } a \geq b \\ -\infty & \text{otherwise} \end{cases} \tag{9}$$

3.4 Learning and Optimization

The goal of BotDGT is to capture both the topological structure and the dynamic nature of social networks to classify the accounts into legitimate user and social bots. We pass the output of the temporal module into a linear layer and softmax layer for bot detection:

$$\hat{\boldsymbol{y}}_i = softmax(\boldsymbol{W}_2 \cdot (\sigma(\boldsymbol{W}_1 \cdot \hat{\boldsymbol{z}}_i + \boldsymbol{b}_1)) + \boldsymbol{b}_2), \tag{10}$$

where $\hat{\boldsymbol{y}}_i$ is the predicted output of node v_i and $\hat{\boldsymbol{z}}_i$ is the representation of node v_i obtained by temporal module. Finally, we define the objective function that utilizes a binary cross-entropy function to classify node v into legitimate users and social bots at each snapshot:

$$Loss = \sum_{k=1}^{N} \sum_{v_i \in V^{t_k}} \left[\boldsymbol{y}_i log(\hat{\boldsymbol{y}}_i) + (1-\boldsymbol{y}_i) log(1-\hat{\boldsymbol{y}}_i)\right], \tag{11}$$

where N is the number of the snapshots, $\boldsymbol{y}_i$ is the ground truth label of node v_i.

4 Experiments

In this section, we conduct extensive experiments on two benchmark datasets to answer the following questions:

- **RQ1:** How does our framework perform in bot detection compared to baseline methods?
- **RQ2:** What is the impact of removing individual architectural components on the framework's performance?
- **RQ3:** What is the significance of capturing the dynamic nature of social networks for social bot detection?

4.1 Experimental Setup

Dataset

We conduct experiments on two comprehensive social bot detection benchmarks: TwiBot-20 [Feng *et al.*, 2021a] and TwiBot-22 [Feng *et al.*, 2022b]. The datasets provide a wide range of entities and relationships, spanning the period from the inception of Twitter to the time of dataset creation, which supports our bot detection framework to model the topological structure and dynamic nature of social networks.

Baselines

We compare BotDGT with comprehensive social bot detection methods categorized into two groups: feature-based methods and graph-based methods. Our code is publicly available on GitHub[1].

Feature-based methods generally extract the numerical features from user metadata or semantic features from textual information to identify social bots, including:

- **EvolveBot** [Yang *et al.*, 2013] designs robust features that are expensive for bots to evade and utilizes machine learning classifiers to combat the evasion tactics of spammers.
- **Varol *et al.*** [Varol *et al.*, 2017] extracts groups of features from Twitter users and leverages random forest classifier to identify Twitter bot.
- **BotBuster** [Ng and Carley, 2023] employs a mixture-of-experts approach to process user metadata and textual information, thereby improving cross-platform bot detection.
- **DeeProBot** [Hayawi *et al.*, 2022] extracts features from the user account and leverages natural language processing techniques to learn user representations for bot detection.
- **SGBot** [Yang *et al.*, 2020] is proposed to tackle the scalability and generalization issues in social bot detection by strategically selecting a subset of training data.

Graph-based methods generally interpret social networks as graphs and leverage geometric deep learning for social bot detection, including:

- **GCN** [Kipf and Welling, 2016] equally aggregates features from neighbors and learns user representations, which are then passed to a linear layer for classification.
- **GAT** [Veličković *et al.*, 2017] leverages an attention mechanism to assign diverse weights to different neighboring nodes, improving the learning of node representations.
- **BotRGCN** [Feng *et al.*, 2021b] is designed to construct a heterogeneous social network graph and employ relational graph convolutional networks for bot detection.
- **RGT** [Feng *et al.*, 2022a] utilizes graph transformers and semantic attention to effectively model the heterogeneity of social networks for bot detection.

4.2 Framework Performance (RQ1)

We evaluate our proposed social bot detection framework along with several representative baselines on the two benchmarks and present the results in Table 1. It is demonstrated

[1]https://github.com/Peien429/BotDGT

Methods	Dataset	TwiBot-20				TwiBot-22			
	Metrics	Accuracy	F1-score	Precision	Recall	Accuracy	F1-score	Precision	Recall
Feature-based	EvolveBot	65.83±0.63	69.75±0.50	66.93±0.60	72.81±0.41	71.09±0.03	14.09±0.08	56.38±0.04	8.04±0.05
	Varol *et al.*	78.74±0.55	81.08±0.48	78.04±0.61	84.37±0.67	73.92±0.02	27.54±0.26	75.74±0.31	16.83±0.21
	BotBuster	78.55±0.44	82.12±0.61	79.85±0.74	84.00±0.53	74.33±0.17	52.26±1.82	63.32±1.47	45.64±1.70
	DeeProBot	73.14±0.01	77.05±0.02	71.61±0.01	83.50±0.04	76.50±0.07	24.74±0.08	**80.00±0.27**	14.99±0.05
	SGBot	79.50±0.72	84.15±0.53	75.64±0.70	93.54±0.36	75.53±0.25	37.45±0.24	74.31±0.16	25.42±0.07
Graph-based	GCN	83.51±0.56	84.81±0.42	84.60±1.26	85.05±0.94	77.83±0.96	52.16±3.46	71.83±1.12	45.77±2.25
	GAT	85.04±0.38	86.65±0.61	83.69±1.23	89.94±1.65	78.65±0.19	55.86±1.38	71.24±0.80	46.04±2.17
	BotRGCN	85.83±0.38	87.44±0.42	83.98±0.34	91.20±1.03	78.95±0.26	56.47±1.21	72.38±1.42	46.33±1.74
	RGT	86.53±0.47	87.74±0.62	**90.37±0.64**	77.47±0.43	77.01±0.21	47.25±0.83	72.80±0.76	34.99±0.90
ours	BotDGT	**87.25±0.51**	**88.87±0.55**	84.44±0.39	**94.24±0.37**	**79.33±0.22**	**58.15±0.74**	72.42±0.70	**48.46±0.92**

Table 1: Performance of different social bot detection methods on TwiBot-20 and TwiBot-22. We run each method five times and report the average value as well as the standard deviation. The best and second-best results are highlighted with **bold** and underline.

Ablation Settings	TwiBot-20		TwiBot-22	
	Accuracy	F1-score	Accuracy	F1-score
BotDGT	87.25±0.51	88.87±0.55	79.33±0.22	58.15±0.74
replace SM w/ GCN	86.28±0.35	87.87±0.33	79.28±0.05	57.14±1.11
replace SM w/ GAT	86.48±0.39	87.99±0.42	79.14±0.13	57.23±1.42
replace SM w/ RGCN	86.33±0.55	87.92±0.73	79.17±0.15	57.70±1.76
replace SM w/ RGT	86.22±0.20	87.64±0.24	79.28±0.05	57.24±1.30
w/o TM	85.72±0.21	86.99±0.39	78.64±0.12	55.23±1.17
w/o $\boldsymbol{p}^{AT}$ in TM	86.42±0.40	88.13±0.54	79.23±0.12	56.51±0.54
w/o $\boldsymbol{p}^{LCC}$ in TM	86.25±0.51	87.92±0.53	79.25±0.21	56.77±0.93
w/o $\boldsymbol{p}^{BLR}$ in TM	86.56±0.22	88.08±0.24	79.29±0.15	56.84±0.98

Table 2: Results of ablation study. SM and TM denote the structural module and temporal module, respectively.

that graph-based methods, which treat the social network as graphs, generally outperform feature-based methods. This could be attributed to the feature-based methods are easily circumvented by forging numerical or semantic features. The results underscore the importance of capturing topological structure of social networks for effective bot detection.

Our proposed BotDGT outperforms other static graph-based baseline models, including the state-of-the-art static graph model, in terms of accuracy, recall, and F1-score on both TwiBot-20 and Twibot-22 datasets. In comparison with the architecture of previous static graph-based methods, BotDGT not only leverages the topological structure of the social network but also incorporates a temporal module that captures the dynamic nature of the social network. The superior performance of BotDGT could be attributed to its ability to capture the historical context of social networks and model the behavior patterns of automated bots that may evolve over time to evade detection, enabling better discrimination of social bots disguised as legitimate users by interacting with other users. Further detailed analysis of the dynamicity modeling is provided in Section 4.4.

4.3 Ablation Study (RQ2)

BotDGT comprises a structural module and a temporal module, to generate node representations for social bot detection. In this section, we conduct an ablation study on BotDGT by removing or replacing one specific component at a time to assess its significance. The components validated in this section are the message-passing mechanism in the structural module, the temporal module and the temporal position embeddings within it. Experimental results of these ablation models on TwiBot-20 and TwiBot-22 are shown in Table 3.

Effect of Structural Attention. In the structural module, we propose a scaled dot-product attention mechanism that assigns diverse weights to different neighboring nodes for propagating messages and capturing topological structure in each static snapshot. We replace the structural attention with other static graph-based methods proposed to detect social bots, including GCN, GAT, BotRGCN, and RGT. The observed performance degradation indicates that the scaled dot-product structural attention better captures the underlying topological structure for dynamicity modeling of social network.

Effect of Temporal Module. The temporal module assumes a pivotal role in modeling the dynamicity of social networks and the evolving behavior patterns of social bots. We first assess the overall impact of the temporal module and then evaluate the effect of position embeddings within the temporal module. The variant **w/o TM** removes the temporal module of BotDGT, characterizing the social network as a static graph. As shown in Table 3, the variant's performance experiences significant degradation when compared to the original BotDGT architecture, which indicates the importance of incorporating temporal information for effective bot detection. The temporal position embeddings are proposed to capture the temporal information in the sequence. The variant **w/o $\boldsymbol{p}^{AT}$ in TM** confirms the effectiveness of capturing the ordering information of the time sequence. The variants **w/o $\boldsymbol{p}^{AT}$ in TM** and **w/o $\boldsymbol{p}^{LCC}$ in TM** demonstrate the importance of considering evolving behavior patterns. Overall, these results confirm the crucial role of positional embeddings in effectively modeling the dynamicity of social networks.

4.4 Significance of Dynamicity Modeling (RQ3)

To investigate the impact of exploiting the inherent dynamicity of social networks for bot detection, we enhance several static graph-based baselines with the proposed temporal module and compare their performance before and after this enhancement. The experimental results outlined in Table 3 reveal that the majority of the graph-based baselines with the temporal module integrated exhibit noticeable performance improvement compared with their non-enhanced

Dataset	Metric	Accuracy		F1-score		Precision		Recall	
	Method	Original	Enhanced	Original	Enhanced	Original	Enhanced	Original	Enhanced
TwiBot-20	GCN	83.51±0.56	**86.28±0.35**	84.81±0.42	**87.87±0.33**	84.60±1.26	84.20±0.24	85.05±0.94	**91.88±0.56**
	GAT	85.04±0.38	**86.48±0.39**	86.65±0.61	**87.99±0.42**	83.69±1.23	**84.65±0.56**	89.94±1.65	**91.61±1.25**
	BotRGCN	85.83±0.38	**86.33±0.55**	87.44±0.42	**87.92±0.73**	83.98±0.34	83.95±0.74	91.20±1.03	**91.77±1.02**
	RGT	86.53±0.47	86.22±0.20	87.74±0.62	87.64±0.24	90.37±0.64	84.00±0.26	77.47±0.43	**91.61±0.79**
TwiBot-22	GCN	77.83±0.96	**79.28±0.05**	52.16±3.46	**57.14±1.11**	71.83±1.12	71.10±1.44	45.77±2.25	**47.82±2.16**
	GAT	78.65±0.19	**79.14±0.13**	55.86±1.38	**57.23±1.42**	71.24±0.80	71.45±2.00	46.04±2.17	**48.77±2.87**
	BotRGCN	78.95±0.26	**79.17±0.15**	56.47±1.21	**57.70±1.76**	72.38±1.42	**73.37±1.65**	46.33±1.74	**48.07±3.02**
	RGT	77.01±0.21	**79.28±0.05**	47.25±0.83	**57.24±1.30**	72.80±0.76	**72.99±1.90**	34.99±0.90	**47.17±2.60**

Table 3: Performance comparison between the original static graph-based baselines and the enhanced models with the proposed temporal module. **Bold** indicates the improved model performance.

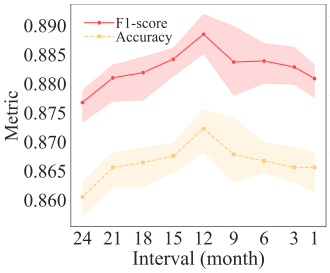

Figure 3: Performance at various time intervals

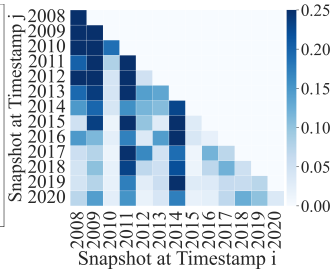

Figure 4: Distribution of attention weights in temporal module

counterparts.

Similar to BotDGT's result, all of the enhanced graph-based methods can achieve higher recall rates, indicating the significance of capturing the dynamic nature of social networks in detecting disguised social bots. Notably, the graph-based methods with the temporal module integrated experience a slight reduction in precision. We speculate that the increased sensitivity to social bots derived from the temporal module might lead to a slight increase in false positives. This observation also gives rise to a decrease in RGT's F1-score on the Twibot-20 dataset, which initially had the highest precision and the lowest recall rate. Nevertheless, there is a notable improvement in F1-score for most graph-based methods, which can reaffirm the importance of incorporating the dynamic nature of social networks in bot detection.

5 Discussion

5.1 Granularity for Dynamicity Modeling

Different from the static graph-based approaches that rely on the most recent snapshot, BotDGT integrates historical context from multiple snapshots. During the process of graph construction in BotDGT, we construct a series of snapshots to depict the social network at a time interval Δ_t, which determines the granularity of the dynamic social network. To explore how granularity affects dynamicity modeling, we evaluate BotDGT with various time intervals on TwiBot-20. The result, illustrated in Figure 3, shows that the model performance initially improves as the granularity becomes finer due to richer temporal information. However, a decline in the model performance is observed when the granularity becomes finer than 12 months. We speculate that while setting the granularity finer than 12 months provides more historical context and detailed evolving patterns, it also introduces more noise and short-term fluctuations of the social network, making it more challenging for the temporal module to learn consistent patterns. Furthermore, the study [Cresci, 2020] has found that the evolution of social bots is not very frequent, indicating that social bots mostly don't change their actions or strategies within a short period of time. Therefore, excessively fine granularity may not provide meaningful insights into the evolving behavior patterns exhibited by social bots.

5.2 Temporal Attention for Dynamicity Modeling

To further explore how temporal attention affects the performance of bot detection, we visualize the distribution of temporal attention weights averaged over test nodes on TwiBot-20 at the time interval of 12 months. In Figure 4, each row represents the attention weight distribution of the snapshot at t_k over its historical snapshots at $t_1, \cdots, t_{k-1}$. As shown in Figure 4, BotDGT doesn't assign uniform weight to historical snapshots, indicating the different contributions of each snapshot for social bot detection. When predicting the most recent snapshot of the dynamic social network, BotDGT assigns more weight to the historical snapshots before 2015, rather than focusing on the more recent ones. We speculate that the reason is that advanced social bots evolved to change their behavior patterns and interact with legitimate users around 2015, which is consistent with *the rise of a third wave of bots* from 2016 onwards as described in the study [Cresci, 2020]. Thus, BotDGT is capable of adapting attention weight distributions to effectively incorporate historical context.

6 Conclusion

The proliferation of social network bots has led to negative consequences. While state-of-the-art bot detection methods generally represent the social network as a static graph, they tend to overlook the dynamic nature of the social network. In this paper, we propose a bot detection framework, BotDGT, to exploit the inherently dynamic nature of social networks, which incorporates the historical context and models the evolution of behavior patterns. Experimental results on real-world datasets demonstrate that BotDGT outperforms other static graph-based bot detection methods. Further studies indicate the significance of exploiting the social network's dynamic nature for effective bot detection.

Acknowledgments

This work was supported by the National Key Research and Development Program of China through grants 2022YFB3104700, 2022YFB3105405, 2021YFC3300502, NSFC through grants 62322202, U2003206, Beijing Natural Science Foundation through grant 4222030, Guangdong Basic and Applied Basic Research Foundation through grant 2023B1515120020, Shijiazhuang Science and Technology Plan Project through grant 231130459A.

References

[Ali Alhosseini *et al.*, 2019] Seyed Ali Alhosseini, Raad Bin Tareaf, Pejman Najafi, and Christoph Meinel. Detect me if you can: Spam bot detection using inductive representation learning. In *WWW*, pages 148–153, 2019.

[Alvarez-Rodriguez *et al.*, 2021] Unai Alvarez-Rodriguez, Federico Battiston, Guilherme Ferraz de Arruda, Yamir Moreno, Matjaž Perc, and Vito Latora. Evolutionary dynamics of higher-order interactions in social networks. *Nature Human Behaviour*, 5(5):586–595, 2021.

[Cai *et al.*, 2021] Lei Cai, Zhengzhang Chen, Chen Luo, Jiaping Gui, Jingchao Ni, Ding Li, and Haifeng Chen. Structural temporal graph neural networks for anomaly detection in dynamic graphs. In *CIKM*, pages 3747–3756, 2021.

[Chen *et al.*, 2022] Jinyin Chen, Xueke Wang, and Xuanheng Xu. Gc-lstm: Graph convolution embedded lstm for dynamic network link prediction. *Applied Intelligence*, pages 1–16, 2022.

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

[Cui *et al.*, 2020] Limeng Cui, Haeseung Seo, Maryam Tabar, Fenglong Ma, Suhang Wang, and Dongwon Lee. Deterrent: Knowledge guided graph attention network for detecting healthcare misinformation. In *SIGKDD*, pages 492–502, 2020.

[Dukić *et al.*, 2020] David Dukić, Dominik Keča, and Dominik Stipić. Are you human? detecting bots on twitter using bert. In *DSAA*, pages 631–636. IEEE, 2020.

[Feng *et al.*, 2021a] Shangbin Feng, Herun Wan, Ningnan Wang, Jundong Li, and Minnan Luo. Twibot-20: A comprehensive twitter bot detection benchmark. In *CIKM*, pages 4485–4494, 2021.

[Feng *et al.*, 2021b] Shangbin Feng, Herun Wan, Ningnan Wang, and Minnan Luo. Botrgcn: Twitter bot detection with relational graph convolutional networks. In *SNAM*, pages 236–239, 2021.

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

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

[Ferrara *et al.*, 2016] Emilio Ferrara, Wen-Qiang Wang, Onur Varol, Alessandro Flammini, and Aram Galstyan. Predicting online extremism, content adopters, and interaction reciprocity. In *Social Informatics: 8th International Conference, SocInfo 2016, Bellevue, WA, USA, November 11-14, 2016, Proceedings, Part II 8*, pages 22–39. Springer, 2016.

[Gao *et al.*, 2023] Yuan Gao, Xiang Wang, Xiangnan He, Huamin Feng, and Yongdong Zhang. Rumor detection with self-supervised learning on texts and social graph. *FCS*, 17(4):174611, 2023.

[Gehring *et al.*, 2017] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *ICML*, pages 1243–1252. PMLR, 2017.

[Gilmer *et al.*, 2017] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *ICML*, pages 1263–1272. PMLR, 2017.

[Guo *et al.*, 2022] Xingzhi Guo, Baojian Zhou, and Steven Skiena. Subset node anomaly tracking over large dynamic graphs. In *SIGKDD*, pages 475–485, 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.

[Kim *et al.*, 2022] Seoyoon Kim, Seongjun Yun, and Jaewoo Kang. Dygrain: An incremental learning framework for dynamic graphs. In *IJCAI*, pages 3157–3163, 2022.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv*, 2016.

[Lee *et al.*, 2011] Kyumin Lee, Brian Eoff, and James Caverlee. Seven months with the devils: A long-term study of content polluters on twitter. In *ICWSM*, volume 5, pages 185–192, 2011.

[Liu *et al.*, 2020] Fanzhen Liu, Jia Wu, Shan Xue, Chuan Zhou, Jian Yang, and Quanzheng Sheng. Detecting the evolving community structure in dynamic social networks. *World Wide Web*, 23:715–733, 2020.

[Mazza *et al.*, 2019] Michele Mazza, Stefano Cresci, Marco Avvenuti, Walter Quattrociocchi, and Maurizio Tesconi. Rtbust: Exploiting temporal patterns for botnet detection on twitter. In *Proceedings of the 10th ACM conference on web science*, pages 183–192, 2019.

[Miller *et al.*, 2014] Zachary Miller, Brian Dickinson, William Deitrick, Wei Hu, and Alex Hai Wang. Twitter spammer detection using data stream clustering. *Information Sciences*, 260:64–73, 2014.

[Ng and Carley, 2023] Lynnette Hui Xian Ng and Kathleen M Carley. Botbuster: Multi-platform bot detection using a mixture of experts. In *ICWSM*, volume 17, pages 686–697, 2023.

[Pareja *et al.*, 2020] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *AAAI*, volume 34, pages 5363–5370, 2020.

[Peng *et al.*, 2021] Hao Peng, Renyu Yang, Zheng Wang, Jianxin Li, Lifang He, S Yu Philip, Albert Y Zomaya, and Rajiv Ranjan. Lime: Low-cost and incremental learning for dynamic heterogeneous information networks. *TC*, 71(3):628–642, 2021.

[Peng *et al.*, 2024] Hao Peng, Jingyun Zhang, Xiang Huang, Zhifeng Hao, Angsheng Li, and Zhengtao Yu. Unsupervised social bot detection via structural information theory. *TOIS*, 2024.

[Pham *et al.*, 2022] Phu Pham, Loan TT Nguyen, Bay Vo, and Unil Yun. Bot2vec: A general approach of intra-community oriented representation learning for bot detection in different types of social networks. *Information Systems*, 103:101771, 2022.

[Rossi *et al.*, 2020] Sippo Rossi, Matti Rossi, Bikesh Upreti, and Yong Liu. Detecting political bots on twitter during the 2019 finnish parliamentary election. 2020.

[Sankar *et al.*, 2020] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *WSDM*, pages 519–527, 2020.

[Shi *et al.*, 2023] Shuhao Shi, Kai Qiao, Jie Yang, Baojie Song, Jian Chen, and Bin Yan. Over-sampling strategy in feature space for graphs based class-imbalanced bot detection. *arXiv*, 2023.

[Skarding *et al.*, 2021] Joakim Skarding, Bogdan Gabrys, and Katarzyna Musial. Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey. *IEEE Access*, 9:79143–79168, 2021.

[Subrahmanian *et al.*, 2016] Venkatramanan S Subrahmanian, Amos Azaria, Skylar Durst, Vadim Kagan, Aram Galstyan, Kristina Lerman, Linhong Zhu, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. The darpa twitter bot challenge. *Computer*, 49(6):38–46, 2016.

[Sun *et al.*, 2021] Li Sun, Zhongbao Zhang, Jiawei Zhang, Feiyang Wang, Hao Peng, Sen Su, and S Yu Philip. Hyperbolic variational graph neural network for modeling dynamic graphs. In *AAAI*, 2021.

[Varol *et al.*, 2017] Onur Varol, Emilio Ferrara, Clayton Davis, Filippo Menczer, and Alessandro Flammini. Online human-bot interactions: Detection, estimation, and characterization. In *ICWSM*, volume 11, 2017.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[Veličković *et al.*, 2017] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv*, 2017.

[Wang *et al.*, 2021] Yanbang Wang, Pan Li, Chongyang Bai, and Jure Leskovec. Tedic: Neural modeling of behavioral patterns in dynamic social interaction networks. In *WWW*, pages 693–705, 2021.

[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 *TPS-ISA*, pages 101–109. IEEE, 2019.

[Wu *et al.*, 2023] Qi Wu, Yingguan Yang, Buyun He, Hao Liu, Xiang Wang, Yong Liao, Renyu Yang, and Pengyuan Zhou. Heterophily-aware social bot detection with supervised contrastive learning. *arXiv*, 2023.

[Xie *et al.*, 2021] Yuanzhen Xie, Zijing Ou, Liang Chen, Yang Liu, Kun Xu, Carl Yang, and Zibin Zheng. Learning and updating node embedding on dynamic heterogeneous information network. In *WSDM*, pages 184–192, 2021.

[Xu *et al.*, 2020] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. *arXiv*, 2020.

[Yang *et al.*, 2013] Chao Yang, Robert Harkreader, and Guofei Gu. Empirical evaluation and new design for fighting evolving twitter spammers. *TIFS*, 8(8):1280–1293, 2013.

[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 *AAAI*, volume 34, pages 1096–1103, 2020.

[Yang *et al.*, 2023a] Yingguang Yang, Renyu Yang, Yangyang Li, Kai Cui, Zhiqin Yang, Yue Wang, Jie Xu, and Haiyong Xie. Rosgas: Adaptive social bot detection with reinforced self-supervised gnn architecture search. *TWEB*, 17(3):1–31, 2023.

[Yang *et al.*, 2023b] Yingguang Yang, Renyu Yang, Hao Peng, Yangyang Li, Tong Li, Yong Liao, and Pengyuan Zhou. Fedack: Federated adversarial contrastive knowledge distillation for cross-lingual and cross-model social bot detection. In *WWW*, pages 1314–1323, 2023.

[Ying *et al.*, 2021] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *NeurIPS*, 34:28877–28888, 2021.

[Zeng *et al.*, 2024] Xianghua Zeng, Hao Peng, and Angsheng Li. Adversarial socialbots modeling based on structural information principles. In *AAAI*, volume 38, pages 392–400, 2024.

[Zhang *et al.*, 2023] Guozhen Zhang, Tian Ye, Depeng Jin, and Yong Li. An attentional multi-scale co-evolving model for dynamic link prediction. In *WWW*, pages 429–437, 2023.

[Zhou *et al.*, 2020] Fan Zhou, Xovee Xu, Ce Li, Goce Trajcevski, Ting Zhong, and Kunpeng Zhang. A heterogeneous dynamical graph neural networks approach to quantify scientific impact. *arXiv*, 2020.