BIC: Twitter Bot Detection with Text-Graph Interaction and Semantic Consistency

Anonymous ACL submission

Abstract

Twitter bots are automatic programs operated by malicious actors to manipulate public opinion and spread misinformation. Research efforts have been made to automatically identify bots based on texts and networks on social media. Existing methods only leverage texts or networks alone, and while few works explored the shallow combination of the two modalities, we hypothesize that the interaction and information exchange between texts and graphs could be crucial for holistically evaluating bot activities on social media. In addition, according to a recent survey (Cresci, 2020), Twitter bots are constantly evolving while advanced bots steal genuine users' tweets and dilute their malicious content to evade detection. This results in greater inconsistency across the timeline of novel Twitter bots, which warrants more attention. In light of these challenges, we propose **BIC**, a Twitter **B**ot detection framework with text-graph **I**nteraction and semantic **C**onsistency. Specifically, in addition to separately modeling the two modalities on social media, BIC employs a text-graph interaction module to enable information exchange across modalities in the learning process. In addition, given the stealing behavior of novel Twitter bots, BIC proposes to model semantic consistency in tweets based on attention weights while using it to augment the decision process. Extensive experiments demonstrate that BIC consistently outperforms state-of-the-art baselines on two widely adopted datasets. Further analyses reveal that text-graph interactions and modeling semantic consistency are essential improvements and help combat bot evolution.

1 Introduction

Twitter bots are controlled by automated programs and manipulated to pursue malicious goals such as advocating for extremism and producing spam (Dickerson et al., 2014; Berger and Morgan, 2015). Bots are also involved in spreading misinformation during the pandemic (Shi et al., 2020).

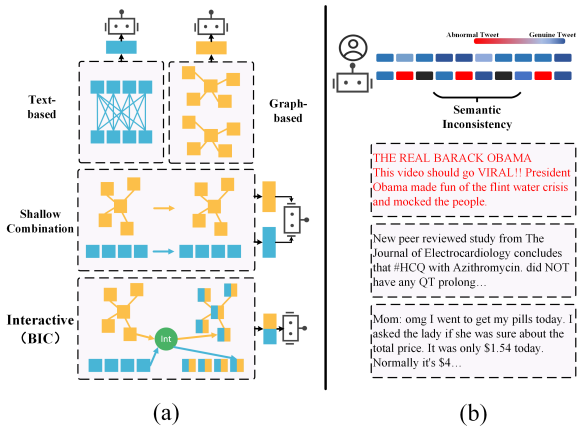

Figure 1: (a) Different types of combining modalities. Previous methods adopt text modality and graph modality alone, or just shallow combine them. There is a need for an interactive method that interacts and exchanges information across the modalities. (b) Genuine users and Twitter bots have different patterns of semantic consistency. Tweets in red are abnormal and these example tweets show semantic inconsistency.

Since Twitter bots pose threat to online society, many efforts have been devoted to detecting bots.

The majority of the existing approaches are text-based and graph-based. The text-based methods analyze the content to detect Twitter bots by natural language processing techniques. Kudugunta and Ferrara (2018) adopted recurrent neural networks to extract textual information. Guo et al. (2021) utilized the pre-trained language model BERT to help detect bots. The graph-based methods model the Twittersphere as graphs and adopt geometric neural networks or concepts of network dynamics to identify bots. (Feng et al., 2022a) constructed a heterogeneous graph and leveraged the different relation information. Magelinski et al. (2020a) exploited the ego-graph of Twitter users and proposed a histogram and customized backward operator.

However, existing methods are faced with two challenges. On the one hand, these methods only adopt texts or graphs alone, and only a few works

shallowly combine the two modalities as Figure 1(a) shows. The text-based model can not get the graph modality information while the graph-based model can not get the text modality information. We hypothesize that it is wise to interact and exchange information between texts and graphs to evaluate bot activities. On the other hand, Cresci (2020) pointed out that Twitter bots are constantly evolving. Advanced bots steal genuine users' tweets and dilute their malicious content to evade detection, which results in greater inconsistency across the timeline of advanced bots as Figure 1(b) illustrates. Previous methods can not capture this characteristic. Namely, there is an urgent need for a method that can identify advanced bots.

In inspire of these challenges, we propose a framework BIC (Twitter **B**ot Detection with Text-Graph **I**nteraction and Semantic **C**onsistency). BIC separately models the two modalities, text and graph, in social media. A text module is adopted to encode the textual information and a graph module to encode graph information. BIC employs a text-graph interaction module to enable different modality information exchange across modalities in the learning process. To capture the inconsistency of advanced bots, BIC leverages a semantic consistency module, which employs the attention weights and a sample pooling function. Our main contributions are summarized as follows:

- We propose to interact and exchange information across text and graph modalities to help detect bots. We find that capturing novel bots' inconsistency can increase detection performance.
- We propose a novel Twitter bot detection model, BIC. It is an end-to-end model and contains a text-graph interaction module to exchange modality information and a semantic consistency module to capture the inconsistency of advanced bots.
- We conduct extensive experiments to evaluate BIC and state-of-the-art models on two widely used datasets. Results illustrate that BIC outperforms all baseline methods. Further analyses reveal the effectiveness of the text-graph interaction module and semantic consistency module.

2 Problem Definition

We first define the task of Twitter bot detection with the text and graph modality. For a Twitter user $u_i \in U$, the text modality contains the description B_i and the tweets $S_i = \{S_{i,j}\}_{j=1}^{T_i}$, where T_i denotes the tweet count. The graph modality contains the representation f_i of u_i and the heterogeneous graph $\mathcal{G} = \mathcal{G}(U, E, \varphi, R^e)$, where U denotes the user set, E denotes the edge set, $\varphi : E \to R^e$ denotes the relation mapping function and R^e is the relation type set. The neighbors of u_i can be derived from $\mathcal{G}$ as $N_i = \{n_{i,j}\}_{j=1}^{J_i}$ where J_i is the neighbor count. The goal is to find a detection function $f : f(u_i) \to \hat{y} \in \{0, 1\}$, such that $\hat{y}$ approximates ground truth y to maximize prediction accuracy.

3 Methodology

Figure 2 displays an overview of our proposed framework named BIC. Specifically, BIC firstly leverages a text module to encode textual information and a graph module to encode graph information. BIC then adopts a text-graph interaction module to interact and exchange modality information in the learning process. To further interact the two modalities, BIC repeats this process for M times. BIC extracts the semantic consistency from the attention weights from the text module with the help of the semantic consistency module. Finally, BIC leverages text modality, graph modality, and semantic consistency vectors to identify bots.

3.1 Modality Interaction

For simplicity, we omit the subscript of the user. BIC first encodes the text modality and graph modality information to obtain the initial representations. For text modality, BIC employs pre-trained RoBERTa (Liu et al., 2019) to encode description B and tweets $\{S\}_{i=1}^{T}$ into $h_{int}^{(0)}$ and $\{h_i^{(0)}\}_{i=1}^{T}$. BIC considers $h_{int}^{(0)}$ as the text interaction modality because the description generally defines the user. For graph modality, BIC employs the same encoding methods as BotRGCN (Feng et al., 2021c) to get the graph interaction representation $g_{int}^{(0)}$ and representations of its neighbors $\{g_i^{(0)}\}_{i=1}^{J}$.

After obtaining the initial representations, BIC employs M times modality interaction to ensure text and graph information interact completely. We describe the l-th interact process as follows.

Text Module BIC puts text representations into a language model to extract textual information, *i.e.*,

$$\{\tilde{h}_{int}^{(l)}, \tilde{h}_1^{(l)}, \cdots, \tilde{h}_T^{(l)}\} = \mathrm{LM}(\{h_{int}^{(l-1)}, h_1^{(l-1)}, \cdots, h_T^{(l-1)}\}), \quad (1)$$

where $\tilde{h}_{int}^{(l)}$ denotes interaction representation of text modality before interaction. BIC adopts trans-

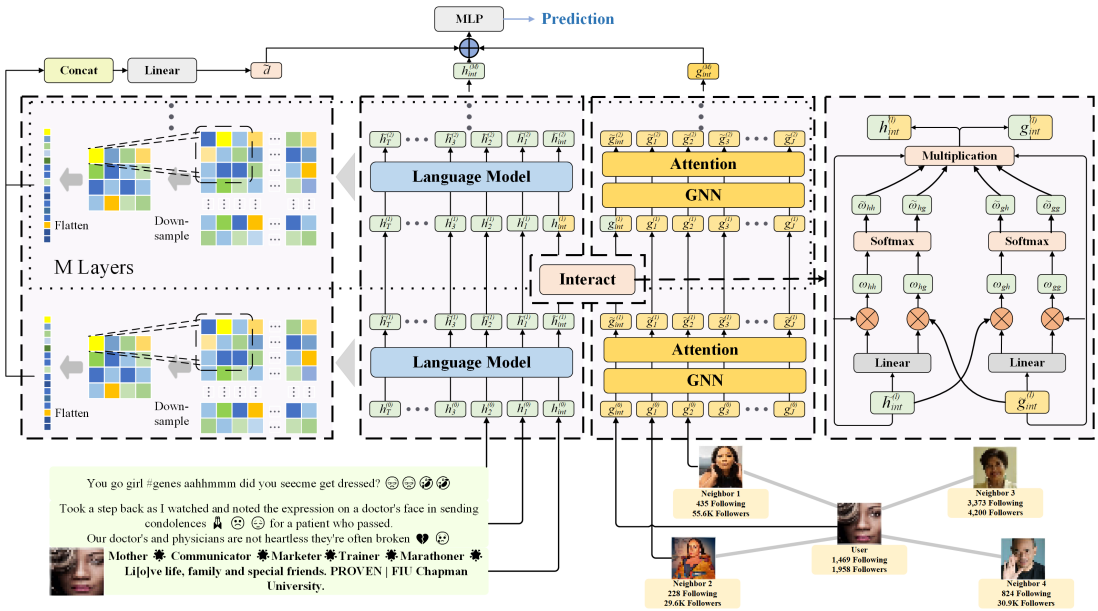

Figure 2: Overview of our proposed framework BIC.

former with multi-head attention (Vaswani et al., 2017) as the language model LM.

graph module BIC firstly feeds graph representations into a graph neural network to aggregate information between users and its neighbors, *i.e.*,

$$\{\hat{g}_{int}^{(l)}, \hat{g}_1^{(l)}, \cdots, \hat{g}_J^{(l)}\} = \mathrm{GNN}(\{g_{int}^{(l-1)}, g_1^{(l-1)}, \cdots, g_J^{(l-1)}\}).$$

BIC adopts relational graph convolutional networks (Schlichtkrull et al., 2018) due to its ability to extract heterogeneous information. To measure which neighbor is important for bot detection, BIC employs multi-head attention for the user, *i.e.*,

$$\{\tilde{g}_{int}^{(l)}, \tilde{g}_1^{(l)}, \cdots, \tilde{g}_J^{(l)}\} = \mathrm{att}(\{\hat{g}_{int}^{(l)}, \hat{g}_1^{(l)}, \cdots, \hat{g}_J^{(l)}\}),$$

where $\tilde{g}_{int}^{(l)}$ denotes interaction representation of graph modality before interaction and att denotes multi-head attention.

3.1.1 Text-Graph Interaction Module

BIC adopts a text-graph interaction module to interact and exchange information across text and graph modality in the learning process. specifically, BIC employ a interaction function inter to interact the text modality representation $\tilde{h}_{int}^{(l)}$ and the graph modality representation $\tilde{g}_{int}^{(l)}$, *i.e.*,

$$(g_{int}^{(l)}, h_{int}^{(l)}) = \mathrm{inter}(\tilde{g}_{int}^{(l)}, \tilde{h}_{int}^{(l)}).$$

For the detail about inter function, BIC calculates the similarity coefficient between modality representations, *i.e.*,

$$\begin{aligned} w_{hh} &= \tilde{h}_{int}^{(l)} \otimes (\theta_1 \cdot \tilde{h}_{int}^{(l)}), \\ w_{hg} &= \tilde{h}_{int}^{(l)} \otimes (\theta_2 \cdot \tilde{g}_{int}^{(l)}), \\ w_{gg} &= \tilde{g}_{int}^{(l)} \otimes (\theta_2 \cdot \tilde{g}_{int}^{(l)}), \\ w_{gh} &= \tilde{g}_{int}^{(l)} \otimes (\theta_1 \cdot \tilde{h}_{int}^{(l)}), \end{aligned} \tag{2}$$

where θ_1 and θ_2 are learnable parameters that transform the modality representations into the interaction-sensitive space, and '$\otimes$' denotes the dot product. BIC then applies a softmax function to derive final similarity weights, *i.e.*,

$$\begin{aligned} \tilde{w}_{hh}, \tilde{w}_{hg} &= \mathrm{softmax}(w_{hh}, w_{hg}), \\ \tilde{w}_{gg}, \tilde{w}_{gh} &= \mathrm{softmax}(w_{gg}, w_{gh}). \end{aligned}$$

BIC finally makes the two representations interact through the derived similarity weights, *i.e.*,

$$\begin{aligned} h_{int}^{(l)} &= \tilde{w}_{hh}\tilde{h}_{int}^{(l)} + \tilde{w}_{hg}\tilde{g}_{int}^{(l)}, \\ g_{int}^{(l)} &= \tilde{w}_{gg}\tilde{g}_{int}^{(l)} + \tilde{w}_{gh}\tilde{h}_{int}^{(l)}. \end{aligned}$$

So far, BIC could interact and exchange information across the two modalities.

3.2 Semantic Consistency Detection

Since attention weights from the transformer could indicate the correlations and consistency between tweets, BIC adopts the attention weights to extract the semantic consistency information. BIC can obtain the attention weight matrix $\mathcal{M}_i \in$

$\mathbb{R}^{(T+1)\times(T+1)}$ of text representation from equation (1) in i-th interaction process. BIC then employs a down-sample function to reduce the matrix size and obtain what matters in the matrix, *i.e.*,

$$\tilde{\mathcal{M}}_i = \text{sample}(\mathcal{M}_i),\ \tilde{\mathcal{M}}_i \in \mathbb{R}^{K\times K},$$

where K is a hyperparameter indicating the matrix size. BIC adopts fixed size max-pooling as sample function in the experiments. BIC then flat the matrix and applies a linear transform to obtain the semantic consistency representation, *i.e.*,

$$d_i = \theta_{sc} \cdot \text{Flatten}(\tilde{\mathcal{M}}_i),$$

where θ_{sc} is a shared learnable parameter of each interaction process. Finally, BIC applies an aggregating function to combine the representations of each interaction process, *i.e.*,

$$d = \sigma(W_D \cdot \text{aggr}(\{d_i\}_{i=1}^{M}) + b_D),$$

where W_D and b_D are learnable parameters, σ denotes activate function, and aggr denotes the aggregating function, such as concatenate or mean.

3.3 Training and Inference

BIC concatenates text modality $h_{int}^{(M)}$, graph modality $g_{int}^{(M)}$, and semantic consistency d representation to obtain the representation of a user, *i.e.*,

$$z = W_D \cdot (h_{int}^{(M)} \| g_{int}^{(M)} \| d) + b_D. \quad (3)$$

BIC finally employs a softmax layer to get the predicted probability $\hat{y}$. We adopt cross entropy loss to optimize BIC, *i.e.*,

$$l = -\sum_{i\in U}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] + \lambda \sum_{\omega\in\theta} \omega^2,$$

where U denotes all users in the training set, θ denotes all training parameters, y_i denotes the ground-truth label and λ is a regular coefficient.

4 Experiment

4.1 Experiment Settings

More detailed information about the experiment settings and the implementation details of BIC can be found in the appendix. we submit our code and the best parameters as supplementary materials.

Dataset To evaluate BIC and baselines, we make use of two widely used datasets, Cresci-15 (Cresci et al., 2015) and TwiBot-20 (Feng et al., 2021b). These two datasets provide user follow relationships to support graph-based models. TwiBot-20 includes $229,580$ Twitter users, $33,488,192$ tweets, $33,716,171$ edges while Cresci-15 includes $5,301$ Twitter users, $2,827,757$ tweets, $14,220$ edges.

Baselines We compare BIC with **Botometer** (Davis et al., 2016), **Kudugunta *et al.*** (Kudugunta and Ferrara, 2018), **Wei *et al.*** (Wei and Nguyen, 2019), **Alhosseini *et al.*** (Ali Alhosseini et al., 2019), **BotRGCN** (Feng et al., 2021c), **Yang *et al.*** (Yang et al., 2020), **SATAR** (Feng et al., 2021a), and **RGT** (Feng et al., 2022a).

4.2 Main Results

We first evaluate whether these methods leverage text modality, graph modality, and interact modalities. We then benchmark these baselines on Crescie-15 and TwiBot-20, and present results in Table 1. It is demonstrated that:

- BIC consistently outperforms all baselines including the state-of-art methods RGT (Feng et al., 2022a) with at least 1% improvement of performance on two datasets.

- The methods leveraged graph modality such as RGT (Feng et al., 2022a) generally outperform other methods that only adopt text modality or other features. SATAR (Feng et al., 2021a) achieves competitive performance with the text modality and the graph modality. BIC further makes these two modalities interact to achieve the best performance.

- We conduct the significance test using the unpaired t-test. The improvement between BIC and the second-best baseline RGT is statistically significant with p-value < 0.005 on Creaci-15 and p-value < 0.0005 on TwiBot-20.

In the following, we first study the role of the two modalities and the interaction module in BIC. We then examine the effectiveness of the semantic consistency module in identifying advanced bots. Next, we evaluate the ability of BIC to detect advanced bots. We finally evaluate a specific bot in the datasets to explore how BIC makes the choice.

Table 1: Bot detection performance on Cresci-15 and TwiBot-20 benchmarks. For each baseline except for Botometer which has fixed results, we run 5 times on the same splits with different random seeds. **Text**, **Graph**, **Modality-Int** respectively denote whether baseline leverages text modality, graph modality and modality interaction. **Bold** and underline indicate the highest and second highest performance. 'BIC w/o Graph' and 'BIC w/o Text' indicate BIC without the Graph Module and without the Text Module. BIC achieves the best performance.

Method	Modalities			Cresci-15		TwiBot-20	
	Text	Graph	Modality-Int	Accuracy	F1-score	Accuracy	F1-score
Yang *et al.*				77.08 (±0.21)	77.91 (±0.11)	81.64 (±0.46)	84.89 (±0.42)
Botometer				57.92	66.90	53.09	55.13
Kudugunta *et al.*	✓			75.33 (±0.13)	75.74 (±0.16)	59.59 (±0.65)	47.26 (±1.35)
Wei *et al.*	✓			96.18 (±1.54)	82.65 (±2.47)	70.23 (±0.10)	53.61 (±0.10)
BotRGCN		✓		96.52 (±0.71)	97.30 (±0.53)	83.27 (±0.57)	85.26 (±0.38)
Alhossini *et al.*		✓		89.57 (±0.60)	92.17 (±0.36)	59.92 (±0.68)	72.09 (±0.54)
RGT		✓		97.15 (±0.32)	97.78 (±0.24)	86.57 (±0.41)	88.01 (±0.41)
SATAR	✓	✓		93.42 (±0.48)	95.05 (±0.34)	84.02 (±0.85)	86.07 (±0.70)
BIC w/o Graph	✓			97.16 (±0.58)	97.80 (±0.46)	85.44 (±0.32)	86.97 (±0.41)
BIC w/o Text		✓		96.86 (±0.52)	97.57 (±0.39)	85.78 (±0.48)	87.25 (±0.57)
BIC	✓	✓	✓	**98.35** (±0.24)	**98.71** (±0.18)	**87.61** (±0.21)	**89.13** (±0.15)

4.3 Text-Graph Interaction Study

Modality Effectiveness Study We remove the text modality representation $h_{int}^{(M)}$ and the graph modality representation $g_{int}^{(M)}$ in equation (3), to evaluate the role of each modality. The results are illustrated in Table 1. We can conclude that: (i) Removing any modality will cause a drop in performance, which illustrates that leveraging and making the two modalities interact can help identify bots. (ii) BIC without graph modality can achieve the second-best performance on Cresci-15. Other ablation settings can achieve competitive performance. It is shown that BIC can derive useful information from one modality and the semantic consistency representation can help identify bots.

BIC adopts text and graph modalities and leverages the text-graph interaction module to make information across the two modalities. To further examine the ability of BIC to extract modality information, we gradually remove part of one modality information and conduct experiments. The results in Fig 3 demonstrate that: (i) Every modality information benefits the performance of bot detection. It suggests that bot detection relies on the text modality and the graph modality information. (ii) BIC could keep the performance with less information of one modality. It illustrates that the interaction module is effective in exchanging information across the modalities.

Interaction Function Study BIC employs an interaction function, which transforms representations into an interaction-sensitive space and learns the similarity weights, to exchange the modality information. Apart from our proposed similarity-based interaction, there are several other interaction functions. We replace this function with other functions such as mean or MLP, to evaluate the effectiveness of our proposed interaction function. We apply the following different interaction functions:

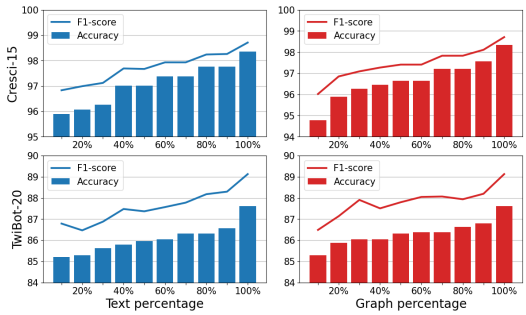

Figure 3: The performance of BIC trained with data that part of one modality is gradually removed. The results illustrate that every modality information benefits the performance and BIC could keep the performance with less information of one modality.

- **Hard** function computes the average of two interaction representations to interact.
- **Soft** function utilizes two learnable parameters as weights for two interaction representations to generate new representations.
- **MLP** function concatenates two interaction representations and feeds the intermediate into an MLP layer to interact.

Table 2: Performance of model with different interaction functions. The results illustrate the effectiveness of the proposed similarity-based interaction.

Function	Cresci-15		TwiBot-20	
	Accuracy	F1-score	Accuracy	F1-score
Ours	**98.35**	**98.71**	**87.61**	**89.13**
w/o interaction	95.89	96.85	85.97	87.42
Hard	96.64	97.41	86.64	88.15
Soft	97.01	97.69	87.06	88.27
MLP	97.38	97.97	86.98	88.44
Text	96.64	97.41	78.53	79.52
Graph	96.45	97.27	86.30	87.65

- **Text** function feeds the interaction representation from text modality into Linear layers.
- **Graph** function feeds the interaction representation from graph modality into Linear layers.

The results in Table 2 illustrate that:

- Almost all interaction strategies outperform methods with no interaction, which indicates the necessity of utilizing an interaction module to make two modalities interactive and exchange information.
- Our similarity-based modality interaction function outperforms others all, which well confirmed its efficacy, indicating that it can truly make two modalities inform each other and learn the relative importance of modalities.

Interaction Number Study To examine the role of the modality information interaction number M, we conduct experiments with different interaction numbers and evaluate the model memory cost. The results in Figure 4 demonstrate that BIC with 2 interactions performs the best over other settings. Besides, the two-interaction model has relatively less memory cost, which makes it the best selection. As the number of interaction number increases, the performance declines gradually, which may be caused by higher complexity increasing the training difficulty. Meanwhile, the one-interaction model may be deficient for learning so rich information, thus leading to unappealing performance.

4.4 Semantic Consistency Study

Discrimination Case Study We check the tweets of users in the used datasets to determine that humans and bots have different semantic consistency patterns and that advanced bots may steal genuine tweets. We choose a genuine user, a traditional bot, and an advanced bot. Their representative tweets are displayed in Figure 5 and we can find that novel bots will have more inconsistency in tweets than genuine users and traditional bots which posts similar rubbish tweets. Next, we check their semantic consistency matrices $\tilde{\mathcal{M}}_i$ and they are shown in Figure 6. We can find that the advanced bot has relatively higher inconsistency in its matrices.

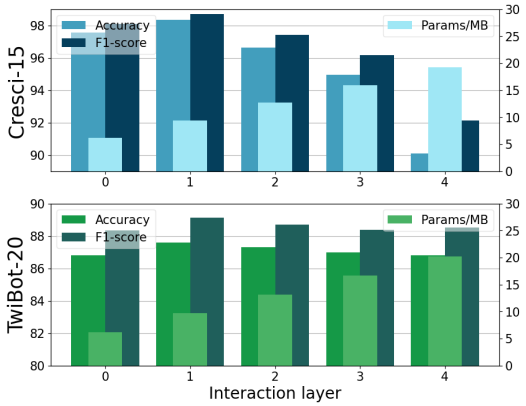

Figure 4: Performance of different numbers of model interaction layers and Params used for one training epoch. The results illustrate that model with 1 interaction layer has good performance with relatively lower Params.

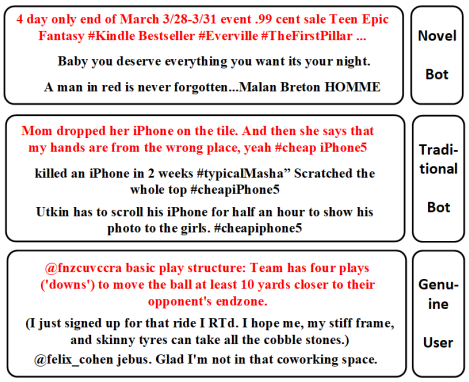

Figure 5: Representative tweets of a genuine user, a traditional bot, and an advanced bot. The tweet in red indicates it has a relatively higher attention weight than other tweets of the same user. More inconsistency has been shown between the advanced bot's tweets in red and tweets in black.

Discrimination Ability Study BIC adopts the attention weight from the text module to generate the semantic consistency representation d. We try to find out the extent to which our semantic consistency module can distinguish bots from genuine users. We derive consistency matrices $\tilde{\mathcal{M}}_i$ and calculate the largest characteristic value. We draw box plots with these characteristic values to find the differences between bots and humans excavated

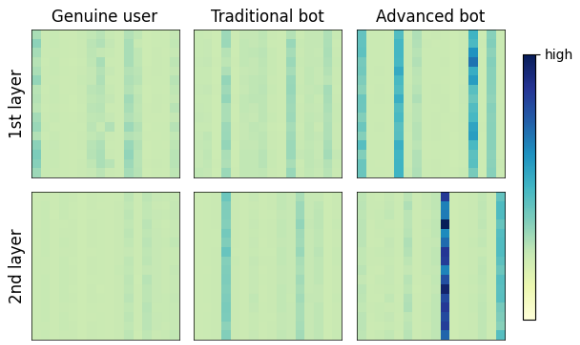

Figure 6: Semantic consistency matrices of a genuine user, a traditional bot, and an advanced bot. The result illustrates that the matrices of advanced bots show more inconsistency than traditional ones or humans.

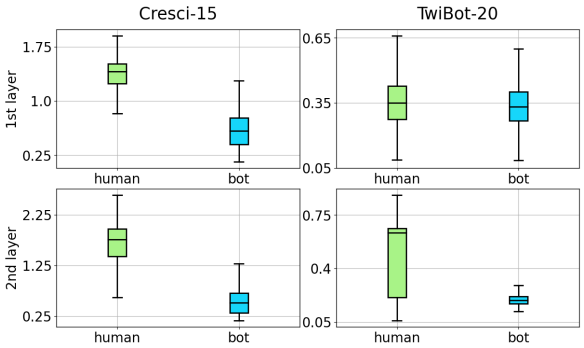

Figure 7: The box plot is drawn from the max characteristic values of the semantic consistency matrices. The results illustrate that the consistency matrices of humans and bots show different patterns.

by the module. The results manifested in Figure 7 demonstrate that the consistency matrices of bots and humans exist in differences.

To evaluate that the semantic consistency representation d can distinguish bots and humans. We conduct the k-means algorithm to cluster the representations and calculate the V-measure, which is a harmonic mean of homogeneity and completeness. BIC achieves 0.4312 of v-measure on Cresci-15 and 0.3336 on TwiBot-20. More intuitively, we adopt t-sne to visualize the representation and the results are shown in Figure 8, which shows moderate collocation for groups of bot and human. It is proven that the semantic consistency representation can identify bots alone.

4.5 Advanced Bot Study

We claim that BIC could identify the advanced bots. To evaluate whether BIC can capture the advanced bots after 2020 (the TwiBot-20 published time), we sample some users related to the pandemic from a

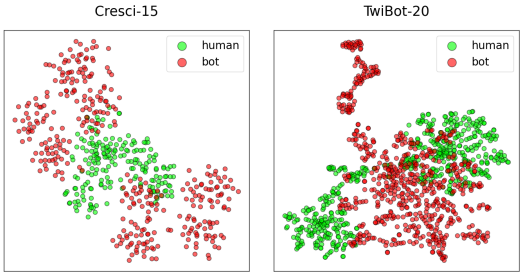

Figure 8: The t-sne plot of the semantic consistency representations. The results illustrate that the representation of humans and the representations of bots are obviously separated, which indicates the effectiveness of the semantic consistency module.

Table 3: Bot detection performance on an up-to-date dataset. BIC outperforms the other two baselines, which illustrates BIC can better identify advanced bots.

Method	Accuracy	F1-score
Botometer	55.35	53.99
RGT	66.95	64.48
BIC	**67.25**	**67.78**

new Twitter crawl (Feng et al., 2022b) to construct a new dataset. This dataset contains user-follow relationships including 5,000 humans and 5,000 bots. We compare BIC with RGT, the second-best baseline, and Botometer, the widely-used bot detection tool. We randomly split this dataset into the train set and the test set by 8:2 and train the methods. Table 3 illustrates the results. We can conclude that BIC achieves the best performance, which proves that BIC can capture advanced bots with the help of the text-graph interaction module and the semantic consistency module.

4.6 Case Study

We study a specific Twitter user to explain how BIC exchanges information across two modalities and learns the relative importance to identify bots. For this user, we study its tweets and neighbors with the top-3 highest attention weight. We then derive similarity weights in Equation (2) to quantitatively analyze it. This user is visualized in Figure 9. We discovered that neighborhood information is more important in this cluster, due to more differences in attention weights of the selected bot's bot neighbors and human neighbors than attention weights of tweets. The conclusion is also reflected in similarity weights. The similarity weights of the original interaction representation from text modality are 0

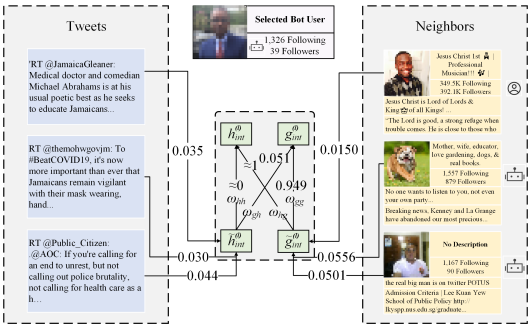

Figure 9: A sample user with its similarity weights inside the box in the middle. On the left are tweets with attention weights from the transformer in the text module. On the right are its neighbors with attention weights from multi-head attention in the graph module.

and 0.051, while the similarity weights of the original interaction representation from graph modality are 1 and 0.949. The results further display the effectiveness of similarity-based interaction in that it indeed learns the emphasis on modalities.

5 Related Work

5.1 Twitter-bot Detection

Text-based Methods Text-based methods adopt techniques in natural language processing to identify bots. Wei and Nguyen (2019) adopted multiple layers of bidirectional LSTM to conduct bot detection. Stanton and Irissappane (2019) proposed to leverage generative adversarial networks to detect spam bots. Hayawi et al. (2022) adopted a variety of features and leveraged LSTM and dense layer to learn representations. Existing models can not capture the semantic consistency of users, which leads to failures to detect the advanced bots.

Graph-based Methods Social network consist of rich information like social familiarity (Dey et al., 2017, 2018), attribution similarity (Peng et al., 2018), and user interaction (Viswanath et al., 2009). The graph constructs on Twittersphere help to detect bots. Feng et al. (2021a) leveraged user neighbor information combined with tweet and profile information. Graph neural networks are utilized to improve the Twitter bot detectors and can achieve great performance (Magelinski et al., 2020b; Dehghan et al., 2022; Yang et al., 2022). Ali Alhosseini et al. (2019) used graph convolutional graph networks to aggregate user information. Feng et al. (2021c) constructed a heterogeneous graph and adopted relational graph convolutional graph networks to identify bots. Previous models leverage the text or graph modality alone without information interaction. We believe that exchanging modality information across two modalities can help improve performance.

5.2 Text-Graph Interaction

Text information is the basis of natural language processing and pre-trained language models are the dominant framework in capturing text features (Devlin et al., 2019; Liu et al., 2019; Lewis et al., 2020). Meanwhile, graph neural networks are introduced to tackle NLP tasks, like fake news detection (Mehta et al., 2022), dialogue state tracking (Feng et al., 2022c), and machine translation (Xu et al., 2021). As both pre-trained LMs and graph structure are proved to be effective, Text-graph interaction was also widely used in the area of natural language processing. Some works interacted with two modalities hierarchically such as using encoded representations from knowledge graph to augment the textual representation (Mihaylov and Frank, 2018; Lin et al., 2019; Yang et al., 2019), or utilizing text representations to enhance the inferential capability of graph (Feng et al., 2020; Lv et al., 2020). More Recently, GreaseLM (Zhang et al., 2022) proposed a model to allow two modalities to interact between layers by interaction nodes, in which truly deep interaction was achieved.

6 Conclusion

Twitter bot detection is a challenging task with increasing importance. To conduct a more comprehensive bot detection, we proposed a bot-detection model named BIC based. BIC interacts and exchanges information across text modality and graph modality by a text-graph interaction module. BIC contains a semantic consistency module that derives the inconsistency from tweets by the attention weight to identify advanced bots. We conducted extensive experiments on two widely used benchmarks to demonstrate the effectiveness of BIC in comparison to competitive baselines. Further experiments also bear out the effectiveness of modality interaction and semantic consistency detection. In the future, we plan to explore better interaction approaches.

References

Seyed Ali Alhosseini, Raad Bin Tareaf, Pejman Najafi, and Christoph Meinel. 2019. Detect me if you can: Spam bot detection using inductive representation

learning. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 148–153.

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

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

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

Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. 2016. Dna-inspired online behavioral modeling and its application to spambot detection. *IEEE Intelligent Systems*, 31(5):58–64.

Clayton Allen Davis, Onur Varol, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. 2016. Botornot: A system to evaluate social bots. In *Proceedings of the 25th international conference companion on world wide web*, pages 273–274.

Ashkan Dehghan, Kinga Siuta, Agata Skorupka, Akshat Dubey, Andrei Betlen, David Miller, Wei Xu, Bogumil Kaminski, and Pawel Pralat. 2022. Detecting bots in social-networks using node and structural embeddings. In *Proceedings of the 11th International Conference on Data Science, Technology and Applications, DATA 2022, Lisbon, Portugal, July 11-13, 2022*, pages 50–61. SCITEPRESS.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Kuntal Dey, Ritvik Shrivastava, Saroj Kaushik, and Kritika Garg. 2018. Assessing topical homophily on twitter. In *International Conference on Complex Networks and their Applications*, pages 367–376. Springer.

Kuntal Dey, Ritvik Shrivastava, Saroj Kaushik, and Vaibhav Mathur. 2017. Assessing the effects of social familiarity and stance similarity in interaction dynamics. In *International Conference on Complex Networks and their Applications*, pages 843–855. Springer.

John P Dickerson, Vadim Kagan, and VS Subrahmanian. 2014. Using sentiment to detect bots on twitter: Are humans more opinionated than bots? In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, pages 620–627. IEEE.

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

Shangbin Feng, Zhaoxuan Tan, Herun Wan, Ningnan Wang, Zilong Chen, Binchi Zhang, Qinghua Zheng, Wenqian Zhang, Zhenyu Lei, Shujie Yang, et al. 2022b. Twibot-22: Towards graph-based twitter bot detection. *arXiv preprint arXiv:2206.04564*.

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

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

Shangbin Feng, Herun Wan, Ningnan Wang, and Minnan Luo. 2021c. 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.

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020. Scalable multi-hop relational reasoning for knowledge-aware question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1295–1309.

Yue Feng, Aldo Lipani, Fanghua Ye, Qiang Zhang, and Emine Yilmaz. 2022c. Dynamic schema graph fusion network for multi-domain dialogue state tracking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 115–126, Dublin, Ireland. Association for Computational Linguistics.

Matthias Fey and Jan Eric Lenssen. 2019. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*.

Qinglang Guo, Haiyong Xie, Yangyang Li, Wen Ma, and Chao Zhang. 2021. Social bots detection via fusing bert and graph convolutional networks. *Symmetry*, 14(1):30.

Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. 2020. Array programming with numpy. *Nature*, 585(7825):357–362.

Kadhim Hayawi, Sujith Mathew, Neethu Venugopal, Mohammad M Masud, and Pin-Han Ho. 2022. Deeprobot: a hybrid deep neural network model for social

bot detection based on user profile data. *Social Network Analysis and Mining*, 12(1):1–19.

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

K Lee, BD Eoff, and J Caverlee. 2011. A long-term study of content polluters on twitter. *ICWSM, seven months with the devils*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2829–2839.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and Songlin Hu. 2020. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8449–8456.

Thomas Magelinski, David Beskow, and Kathleen M Carley. 2020a. Graph-hist: Graph classification from latent feature histograms with application to bot detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5134–5141.

Thomas Magelinski, David Beskow, and Kathleen M Carley. 2020b. Graph-hist: Graph classification from latent feature histograms with application to bot detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5134–5141.

Nikhil Mehta, Maria Leonor Pacheco, and Dan Goldwasser. 2022. Tackling fake news detection by continually improving social context representations using graph neural networks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1363–1380, Dublin, Ireland. Association for Computational Linguistics.

Todor Mihaylov and Anette Frank. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 821–832.

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

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Zhen Peng, Minnan Luo, Jundong Li, Huan Liu, and Qinghua Zheng. 2018. Anomalous: A joint modeling approach for anomaly detection on attributed networks. In *IJCAI*, pages 3513–3519.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.

Wen Shi, Diyi Liu, Jing Yang, Jing Zhang, Sanmei Wen, and Jing Su. 2020. Social bots' sentiment engagement in health emergencies: A topic-based analysis of the covid-19 pandemic discussions on twitter. *International Journal of Environmental Research and Public Health*, 17(22):8701.

Gray Stanton and Athirai Aravazhi Irissappane. 2019. Gans for semi-supervised opinion spam detection. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5204–5210. ijcai.org.

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

Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P. Gummadi. 2009. On the evolution of user interaction in facebook. In *Proceedings of the 2nd ACM Workshop on Online Social Networks*, WOSN '09, page 37–42, New York, NY, USA. Association for Computing Machinery.

Feng Wei and Uyen Trang Nguyen. 2019. Twitter bot detection using bidirectional long short-term memory neural networks and word embeddings. In *2019 First IEEE International Conference on Trust, Privacy*

and Security in Intelligent Systems and Applications (TPS-ISA), pages 101–109. IEEE.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Mingzhou Xu, Liangyou Li, Derek F. Wong, Qun Liu, and Lidia S. Chao. 2021. Document graph for neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8435–8448, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

An Yang, Quan Wang, Jing Liu, Kai Liu, Yajuan Lyu, Hua Wu, Qiaoqiao She, and Sujian Li. 2019. Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2346–2357.

Kai-Cheng Yang, Onur Varol, Pik-Mai Hui, and Filippo Menczer. 2020. Scalable and generalizable social bot detection through data selection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 1096–1103.

Yingguang Yang, Renyu Yang, Yangyang Li, Kai Cui, Zhiqin Yang, Yue Wang, Jie Xu, and Haiyong Xie. 2022. Rosgas: Adaptive social bot detection with reinforced self-supervised gnn architecture search. *arXiv preprint arXiv:2206.06757*.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D. Manning, and Jure Leskovec. 2022. Greaselm: Graph reasoning enhanced language models for question answering. *CoRR*, abs/2201.08860.

A Limitations

The BIC framework has two minor limitations:

- Our proposed BIC model utilizes representation from three different modalities, namely text, graph, and semantic consistency, and we introduce an interaction mechanism to allow information exchange between text and graph. However, whether interaction and information exchange is necessary among all three modalities is still an open question. We leave it to future work to study the necessary by introducing interaction modules.

- The new dataset we construct is limited to the topic of the pandemic while other popular topics are not considered. However, Twitter bots are likely to behave differently with different topics. We leave it to future works to analyze how current approaches perform against bots with different topics.

Table 4: Hyperparameter settings of BIC.

Hyperparameter	**Value**
model layer count M	2
graph module input size	768
graph module hidden size	768
text module input size	768
text module hidden size	768
epoch	30
early stop epoch	10
batch size	64
dropout	0.5
learning rate	1e-4
L2 regularization	1e-5
lr_scheduler_patience	5
lr_scheduler_step	0.1
Optimizer	RAdamW

B Social Impact

Our proposed BIC is a Twitter bot detection model that leverages text-graph interaction and semantic consistency modules. However, there are potential biases or discrimination that exist among the text, graph, or semantic consistency-based representation. For instance, some individuation users may be divided into the bot class for they may behave relevantly 'abnormal'. In conclusion, we suggest that the application of the Twitter bot detection model should be supervised by users and experts.

C Implementation Details

We implement our framework with pytorch (Paszke et al., 2019), PyTorch geometric (Fey and Lenssen, 2019), and the transformer library from huggingface (Wolf et al., 2019). We limit each user's tweet number to 200, and for those who have posted fewer tweets, we bring their initial embeddings up to full strength with vectors made up of all zeros.

C.1 Hyperparamter Setting

Table 4 presents the hyperparameter settings of BIC. For early stopping, we utilize the package provided by Bjarten[1].

[1] https://github.com/Bjarten/early-stopping-pytorch

C.2 Computation

Our proposed method totally has 4.2M learnable parameters and 0.92 FLOPs[2] with hyperparameters presented in Table 4. Our implementation is trained on an NVIDIA GeForce RTX 3090 GPU with 24GB memory, which takes approximately 0.06 GPU hours for training an epoch.

D Baseline Details

- **SATAR** (Feng et al., 2021a) leverages the tweet, profile, and neighbor information and employs a co-influence module to combine them. It pretrains the model with the follower count and finetunes it to detect bots.

- **Botometer** (Davis et al., 2016) is a publicly available service that leverages thousands of features to evaluate how likely a Twitter account exhibits similarity to the known characteristics of typical bots.

- **Kudugunta *et al.*** (Kudugunta and Ferrara, 2018) subdivide bot-detection task to account-level classification and tweet-level classification. In the former, they combine synthetic minority oversampling (SMOTE) with undersampling techniques, and in the latter they propose an architecture that leverages a user's tweets.

- **Wei *et al.*** (Wei and Nguyen, 2019) propose a bot detection model with a three-layer BiLSTM to encode tweets, before which pre-trained GloVe word vectors are used as word embeddings.

- **Alhosseini *et al.*** (Ali Alhosseini et al., 2019) utilize GCN to learn user representations from metadata such as user age, statuses_count, account length name, followers_count to classify bots.

- **BotRGCN** (Feng et al., 2021c) constructs a framework based on relational graph convolutional network (R-GCN) by leveraging representatives derived from the combination of user tweets, descriptions, numerical and categorical property information.

- **Yang *et al.*** (Yang et al., 2020) adopt random forest with account metadata for bot detection, which is proposed to address the scalability and generalization challenge in Twitter bot detection.

- **RGT** (Feng et al., 2022a) leverages relation and influence heterogeneous graph network to conduct bot detection. RGT first learns users' representation under each relation with graph transformers and then integrates representations with the semantic attention network.

[2]https://github.com/Lyken17/pytorch-OpCounter

E Evaluation Details

We elaborate on the evaluation of our baselines here. For methods without text and graph modalities. Lee et al. (2011) adopt random forest classifier with Twitter bot features. Yang et al. (2020) adopt random forest with minimal account metadata. Miller et al. (2014) extract 107 features from a user's tweet and metadata. Cresci et al. (2016) encodes the sequence of a user's online activity with strings. Botometer (Davis et al., 2016) leverages more than one thousand features. All of them extract Twitter bot features, without dealing with these features in graph modality or text modality.

For methods with only text modality, SATAR (Feng et al., 2021a) leverages LSTM for its tweet-semantic sub-network. Kudugunta and Ferrara (2018) adopt deep neural networks for tackling user tweets. Wei and Nguyen (2019) propose a model with a three-layer BiLSTM. All of them deal with user information in text modalities.

For methods with only graph modality, BotRGCN (Feng et al., 2021c) utilizes a relational graph convolutional network in its proposed framework. Ali Alhosseini et al. (2019) adopt graph convolution network to learn user representations and classify bots. RGT (Feng et al., 2022a) leverages heterogeneous graph network to conduct bot detection. All of them deal with user information in graph modalities.

F Semantic consistency Study

Performance study To find how much semantic consistency detection helps the overall BIC performance with different parameter settings, we experiment with different semantic consistency layers, consistency matrix pooling sizes, and consistency vector aggregation manners. The results shown in Fig 10 demonstrate that semantic consistency truly enhances the model performance. Although slight, differences are manifested in different parameter settings, which could be further studied.

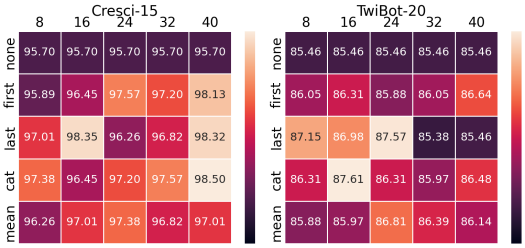

Figure 10: Accuracy of BIC with different settings of considering semantic consistency. The results illustrate the semantic consistency module can improve the performance.

G Scientific Artifact

The BIC model is implemented with the help of many widely-adopted scientific artifacts, including PyTorch (Paszke et al., 2019), NumPy (Harris et al., 2020), transformers (Wolf et al., 2019), sklearn (Pedregosa et al., 2011), PyTorch Geometric (Fey and Lenssen, 2019). We commit to making our code and data publicly available to facilitate reproduction and further research.