Leveraging Cross-Attribute Heterogeneity and Joint Training to Detect Ever-Evolving and Era-Diverse Social Media Bots

Anonymous ACL submission

Abstract

Social media bots detection is a crucial task in maintaining the health of the Internet. The challenge of this task is that bots are evolving themselves by constantly stealing information from human accounts, a behavior also known as feature camouflage, to evade detection. To reduce the impact of camouflage, existing methods detect by using intra-attribute heterogeneity. However, our work reveals that intra-attribute heterogeneity is being diluted by the further stealing behavior of bots, hindering the development of these methods. To address this, we propose a novel concept called cross-attribute heterogeneity. Compared to intra-attribute heterogeneity, it is less susceptible to camouflage. Based on this superior nature, we design a framework called **BCH** to better detect more advanced bots through cross-attribute heterogeneity. Additionally, to enhance compatibility with bots from different eras, BCH incorporates a joint training strategy. Extensive experiments shows the superiority of BCH in detecting ever-evolving and era-diverse bots, as well as detailed analysis highlights the benefits of cross-attribute heterogeneity and the necessity of improving detection methods compatibility.

1 Introduction

Social media has become an integral part of daily life. However, with its growing influence, social media bots have emerged. The bots are a type of special account controlled by automated or semi-automated programs (Davis et al., 2016; Deb et al., 2019; Cresci, 2020). They engage in various malicious activities on social media, severely disrupting the health of the Internet. Therefore, effective bot detection has become a critical research topic.

To better understand the motivation behind our work, it is essential to first grasp the evolution of the bots and how their characteristics change over time. Figure 1(a) shows early primitive bots. The main characteristics of these bots are suspicious features. Suspicious features are caused by bots engaging in malicious activities without any camouflage, and their specific manifestations mainly include an excessive number of statuses, a single topic in posts and a neighbor community composed of bots (Beskow and Carley, 2019; Lee and Kim, 2014; Beskow and Carley, 2018). The emergence of feature camouflage alters the characteristics of primitive bots and transition them into initial camouflage era, as shown in Figure 1(b). During this era, the main characteristic of bots is heterogeneity. The reason why heterogeneity can replace suspicious features is that bots begin to evade detection by stealing information from humans. The original information is covered by the stolen information, thereby reducing suspicious features, while the stolen and original information differ in topics, language styles and other aspects, thereby increasing heterogeneity (Lei et al., 2023; Li et al., 2023). Since this heterogeneity only pertains to a single attribute, it is also referred to as intra-attribute heterogeneity. As camouflage nears completion, bots reshape their characteristic again and evolve into a new era, where they are nearly indistinguishable from humans based on a single attribute, as shown

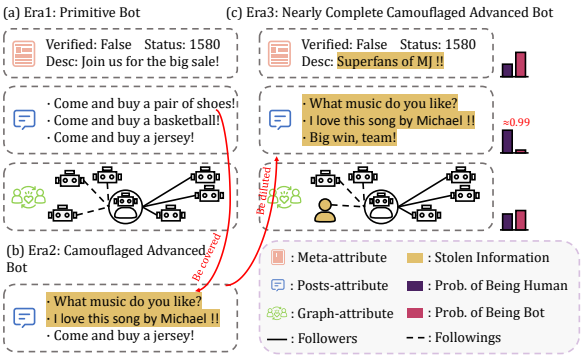

Figure 1: The characteristics of the bots change with their evolution. Meta-attribute provides basic information, posts-attribute introduces posting information and graph-attribute describes neighbor information.

in Figure 1(c). For instance, if all of a bot's posts are stolen from humans, its post-attribute will no longer exhibit any suspicion or heterogeneity.

Existing detection methods fail to exploit the invariant characteristics in bots evolution, putting them at disadvantages in the ongoing battle against evasion strategies. Early methods detect uncamouflaged primitive bots through suspicious features (Yang et al., 2020; Wei and Nguyen, 2019; Feng et al., 2021c,a, 2022a). However, the rise of camouflage reduces the suspicious features, hindering the development of these methods. More recent methods attempt to detect camouflaged bots through intra-attribute heterogeneity (Shi et al., 2023; Ye et al., 2023; Fu et al., 2023; Lei et al., 2023; Liu et al., 2023). However, they still remain vulnerable to nearly complete camouflage, as they focus only on intra-attribute heterogeneity, which also be diluted by bots' persistent stealing behavior.

To address this challenge, our work focuses on the following question: *How can we discover invariant characteristics throughout bots evolution and effectively capture them to detect ever-evolving bots?* Building upon this motivation, we propose **BCH**, a **B**ots detection method with **C**ross-attribute **H**eterogeneity. BCH proposes a novel concept called cross-attribute heterogeneity, as shown in Figure 1(c). The degree of camouflage varies across different attributes, resulting in cross-attribute heterogeneity. This heterogeneity arises from the inherent differences in the cost of camouflaging different attributes. For example, camouflaging graph-attribute requires an entire account as the cost, whereas camouflaging posts-attribute only needs stealing a single post. Therefore, it is bound to exist in the long term and can be regarded as an invariant characteristic. Based on this insight, BCH first encodes different attributes using multiple deep learning models, and then applies attention mechanisms to capture cross-attribute heterogeneity for final detection.

Additionally, another crucial yet overlooked situation is that, while more advanced bots continue to emerge, some residual primitive bots still remain undetected on social media. Therefore, detection efforts should not only be compatible with the future but also with the past. Regrettably, although the heterogeneity can address camouflaged bots, it still struggles with early-era bots, as these primitive bots lack camouflage, leading the absence of heterogeneity in their characteristics. Existing methods fail to realize the limitations of heterogeneity in terms of compatibility with the past. They focus solely on designing increasingly complex frameworks to accommodate heterogeneity but overlooking initial suspicious features, rendering them ineffective in handling era-diverse bots. To overcome this, BCH designs two encoders to separately model suspicious features and heterogeneity. Afterward, BCH use a joint training strategy to adaptively combine these two encoders — when the suspicious features of an account are more pronounced, the corresponding encoder will contribute more to the final detection, and vice versa.

We evaluated our method on three representative datasets, which respectively are dominated by uncamouflaged, camouflaged, and nearly complete camouflaged bots. The experimental results show the superiority of our method in two aspects. First, BCH achieves a 20.76% F1 improvement on dataset dominated by nearly complete camouflaged bots, highlighting the effectiveness of cross-attribute heterogeneity. Second, unlike other methods that excel only on era-specific datasets, BCH performs excellently across all datasets, proving the necessity of the multi-encoder and joint training strategy. Furthermore, detailed analysis reveals the key role of cross-attribute heterogeneity and joint training in detection. Additionally, we explore integrating cross-attribute heterogeneity with large language models, thereby leveraging advanced NLP technique to benefit bots detection task.

Our contributions can be summarized as follows:

- We are the first to leverage evolutionary invariant characteristic for bots detection. It helps us to better handle the ever-evolving bots.
- We are the first to consider the compatibility of detection methods with the past. It helps us to better handle the era-diverse bots.
- We also implement our ideas on advanced large language models, opening up potential directions for future research.

2 Problem Formulation

Given a user U, it consists of meta-attribute M, posts-attribute T and graph-attribute G. M introduces the user's basic information, including boolean, numerical and textual information, T includes the posts generated by the user, and G involves the meta-attribute of the user's followers and followings. Table 1 shows the details of these attributes. Our goal is to detect whether a user is human or bot based on these attributes.

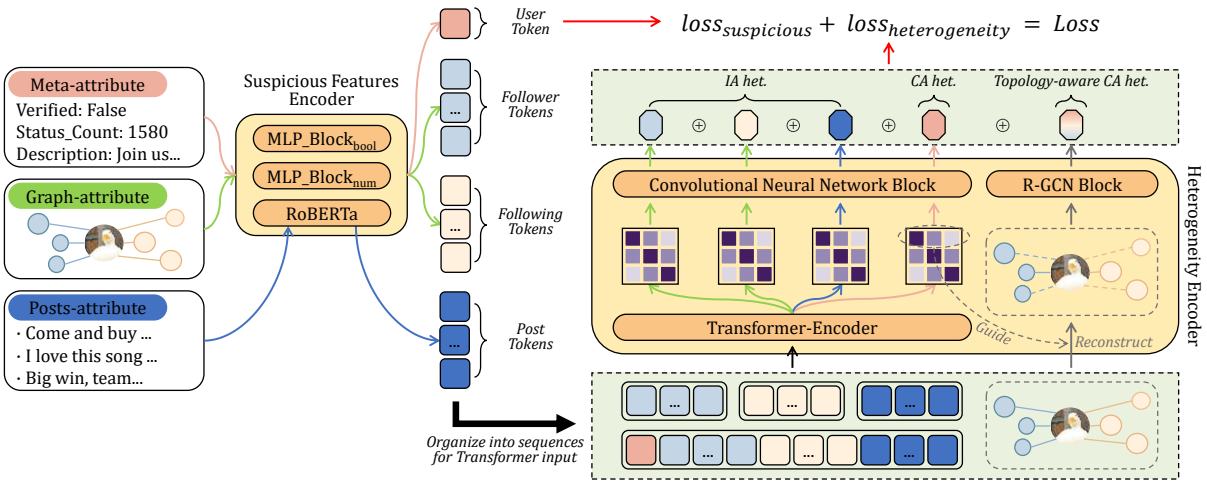

Figure 2: An overall architecture of BCH. In the first step, the suspicious features encoder models suspicious features exhibited by users and tokenizes users all three attributes. In the second step, the heterogeneity encoder captures intra-attribute, cross-attribute and topology-aware cross-attribute heterogeneity. Finally, BCH employs both suspicious features and heterogeneity for detection to combat ever-evolving and era-diverse bots.

3 Methodology

Figure 2 presents an overview of BCH. It comprises a suspicious features encoder and a heterogeneity encoder. These two encoders operate during their respective steps and are integrated by joint training. §A.4 shows the details of hyperparameters involved in this section, such as matrix shapes and more.

3.1 Suspicious Features Encoder

Suspicious features encoder serves two purposes. The primary one is to model suspicious features of users, thereby helping the detection of primitive bots, and the secondary one is to tokenize each attribute of users, thereby facilitating the capture of heterogeneity in subsequent processes.

Regarding the first purpose, this encoder begins by using two MLP blocks and the RoBERTa (Liu et al., 2019) to embed the boolean, numerical and textual information in users meta-attributes. The corresponding embedding results are denoted as r_b, r_v and r_p. Next, this encoder combines all embeddings to construct the suspicious features embedding S. Notably, we do not use users graph-attribute and posts-attribute here, as (Feng et al., 2021c) demonstrate that leveraging meta-attribute alone is sufficient to handle most primitive bots.

Regarding the second purpose, this encoder needs to separately tokenize users all three distinct attributes. For meta-attribute, the suspicious features embedding S can be directly used as the tokenized result. For graph-attribute, since it is composed of the meta-attributes from users followers and followings, it can be tokenized in the same way as the previous step. The tokenized results are respectively denoted as $\{S^i_{fr}\}^{|fr|}_{i=1}$ and $\{S^i_{fw}\}^{|fw|}_{i=1}$. For posts-attribute, it can be tokenized using the RoBERTa of this encoder. The tokenized result is denoted as $\{r^i_t\}^{|T|}_{i=1}$. Both S, $\{S^i_{fr}\}^{|fr|}_{i=1}$, $\{S^i_{fw}\}^{|fw|}_{i=1}$ and $\{r^i_t\}^{|T|}_{i=1}$ will be forwarded as tokens to the next encoder for heterogeneity capturing.

Symbol	Type	Description	Example
M	Tuple	Meta-attribute	(B,V,P)
B	List	Boolean information	Information such as unverified and protected can be represented as [0,1]
V	List	Numerical information	Information such as 12 followers and 27 followings can be represented as [12,27]
P	String	Textual informatin	Information such as user's profile
G	Tuple	Graph-attribute	(fr,fw)
fr	List	Meta-attribute of followers	$\{M^i_{fr}\}^{\|fr\|}_{i=1}$
fw	List	Meta-attribute of followings	$\{M^i_{fw}\}^{\|fw\|}_{i=1}$
T	List	Posts-attribute	$\{t_i\}^{\|T\|}_{i=1}$
t	String	Post	A post generated by a user

Table 1: The details of user's attributes.

3.2 Heterogeneity Encoder

To capture heterogeneity of users, BCH includes a heterogeneity encoder. The structure of this encoder can be roughly divided into three branches. The workflows of the first two branches are similar, they are respectively designed to capture intra-attribute heterogeneity (*IA het.*) and cross-attribute heterogeneity (*CA het.*), and the last branch is tasked with incorporating cross-attribute heterogeneity and topological information.

Branch for Capturing IA Het. In this branch, BCH first organizes the tokens from the previous step into attribute-related sequences, including the follower sequence, following sequence and posts sequence. It then feeds them into a transformer-encoder to capture intra-attribute heterogeneity by computing their self-attention weights. The above process can be represented by E.q (1):

$$\begin{aligned} h_{fr} &= Transformer(\{S_{fr}^{i}\}_{i=1}^{|fr|}), \\ h_{fw} &= Transformer(\{S_{fw}^{i}\}_{i=1}^{|fw|}), \\ h_{t} &= Transformer(\{t_{i}\}_{i=1}^{|T|}). \end{aligned} \tag{1}$$

Afterward, BCH employs a CNN block to downsample these weights, and then flattens the downsampled results to obtain the final intra-attribute heterogeneity embeddings. The corresponding embedding results are denoted as $\tilde{h}_{fr}$, $\tilde{h}_{fw}$ and $\tilde{h}_{t}$.

Branch for Capturing CA Het. The workflow of this branch is similar to capturing intra-attribute heterogeneity, with the main difference being the input to the transformer-encoder. Specifically, in this branch, BCH concatenates the user token, follower tokens, following tokens and posts tokens into a single cross-attribute sequence as the input, and then captures cross-attribute heterogeneity through the same operations as in the previous branch. The obtained cross-attribute heterogeneity attention weight and embedding are respectively denoted as h_c and $\tilde{h}_c$.

Branch for Capturing Topology-aware CA Het. Social media bots often carry out malicious activities in the form of groups, and become neighbors through follower or following relationships (Cresci, 2020). This phenomenon can be described by a topological structure, where nodes represent users and edges represent follower or following relationships. In the branch of capturing cross-attribute heterogeneity, BCH adopts sequential model (Transformer) to process follower and following tokens, resulting in its inability to use the topological information formed by the user and its neighbors. To address this, BCH designed a branch that captures topology-aware cross-attribute heterogeneity.

Specifically, in this branch, BCH first uses the obtained cross-attribute heterogeneity attention weight as a guide to reconstruct the topological information, as shown in E.q (2):

$$\begin{aligned} g &= h_c[0][1{:}\ 1+|fr|+|fw|], \\ hom_{fr}, het_{fr} &= Split(g[:|fr|], k), \\ hom_{fw}, het_{fw} &= Split(g[|fr|:], k), \end{aligned} \tag{2}$$

where the $Split$ function is defined as:

$$\begin{aligned} (L_1, L_2) &= Split(v, k), \\ L_1 &= \{i \mid v_i \in \text{ the largest } \lceil k \times |v| \rceil \text{ of } v\}, \\ L_2 &= \{j \mid j \notin L_1\}, \end{aligned} \tag{3}$$

and $k \in (0, 1]$ is a ratio hyperparameter, a smaller k indicates higher sensitivity to heterogeneity.

Analyzing the results of the $Split$ function, hom_{fr} and hom_{fw} represent neighbors who are assigned higher attention by the user, thus these neighbors can be considered homogeneous with the user. Conversely, het_{fr} and het_{fw} denotes neighbors who are assigned lower attention, and these neighbors can be regarded as heterogeneous with the user. For instance, the heterogeneous neighbors may arise when a bot intentionally following a human account to camouflage itself. Therefore, in summary, BCH achieves a integration of cross-attribute heterogeneity and topological information through the reconstruction.

BCH then adopts a R-GCN (Schlichtkrull et al., 2018) block, to embed the reconstructed topological information, and finally extracts current users embedding from topological information to represent topology-aware cross-attribute heterogeneity. The above process can be represented by E.q (4):

$$\begin{aligned} h_{tc} &= RGCN\_Block(\mathcal{G}), \\ \tilde{h}_{tc} &= h_{tc}[0], \end{aligned} \tag{4}$$

where the $\mathcal{G}$ is defined as:

$$\begin{aligned} \mathcal{V} &= \{S\} \cup \{S_{fr}^{i}\}_{i=1}^{|fr|} \cup \{S_{fw}^{i}\}_{i=1}^{|fw|}, \\ \mathcal{E} &= \{< S, v_j, r_j > \mid v_j \in \mathcal{V} \backslash \{S\}, r_j \in \mathcal{R}\}, \\ \mathcal{G} &= (\mathcal{V}, \mathcal{E}), \end{aligned} \tag{5}$$

and $\mathcal{R} = \{0, 1, 2, 3\}$ respectively represents the four reconstructed relationships, including homogeneous follower, homogeneous following, heterogeneous follower and heterogeneous following.

3.3 Joint Training Strategy

The encoders of BCH provide two different groups of bots' characteristics. The embedding S from suspicious features encoder provide suspicious features for detecting uncamouflaged bots, and the embeddings $\tilde{h}_{fr}$, $\tilde{h}_{fw}$, $\tilde{h}_t$, $\tilde{h}_c$ and $\tilde{h}_{tc}$ from heterogeneity encoder provide heterogeneity for detecting camouflaged bots. To adaptively leverage different characteristics, BCH designs a confidence calculation formula, as shown in E.q (6):

$$\alpha = \frac{H(\hat{y}_s)^{-1}}{H(\hat{y}_s)^{-1} + H(\hat{y}_h)^{-1}} \tag{6}$$

where $H(\hat{y}_*)$ denotes the entropy function:

$$H(\hat{y}_*) = -\sum_{i=0}^{1} p(\hat{y}_*^{(i)}) \log p(\hat{y}_*^{(i)}), \tag{7}$$

and $\hat{y}_s$ and $\hat{y}_h$ are prediction vectors generated by employing softmax to each group of embeddings.

The confidence $\alpha \in [0, 1]$. When BCH is more certain about the judgment based on suspicious features (*i.e.*, when $|\hat{y}_s^{(0)} - \hat{y}_s^{(1)}|$ is larger), α increases. Conversely, when BCH is more certain about the heterogeneity, α decreases. Therefore, BCH treats it as a weighting coefficient to construct final loss function, as shown in E.q (8):

$$\mathcal{L} = \alpha \cdot \mathcal{L}_s + (1 - \alpha) \cdot \mathcal{L}_h, \tag{8}$$

where $\mathcal{L}_s$ and $\mathcal{L}_h$ denotes losses from y_s and y_h.

4 Experiment

4.1 Experimental Setup

Datasets We claim that BCH can address ever-evolving and era-diverse bots effectively. Therefore, we want to select three different types of datasets, where the first consists of nearly complete camouflaged bots, the second consists of initial camouflaged bots and the third consists of uncamouflaged bots. Using datasets with such nature will make our experimental results more convincing.

To achieve this, we propose a LLM-based bot type identification method, as shown in §A.1. We apply it to multiple widely used datasets and ultimately select the three most representative ones. The chosen datasets include TwiBot-22 (Feng et al., 2022b), dominated by nearly fully camouflaged bots; TwiBot-20 (Feng et al., 2021b), dominated by initial camouflaged bots; and Cresci-15 (Cresci et al., 2015), dominated by uncamouflaged bots. §A.1 presents more details of these datasets.

Baselines We select eleven advanced baselines for comparison, which include CACL(Chen et al., 2024), LMBot (Cai et al., 2024), BIC (Lei et al., 2023), BotPercent (Tan et al., 2023), BotMoE (Liu et al., 2023), Hays *et al* (Hays et al., 2023), RGT (Feng et al., 2022a), BotRGCN (Feng et al., 2021c), SGBot (Yang et al., 2020), Alhosseini *et al* (Alhosseini et al., 2019), Wei *et al* (Wei and Nguyen, 2019). They cover methods for detecting early-era primitive bots by modeling suspicious features, as well as methods for detecting camouflaged bots by capturing intra-attribute heterogeneity. §A.2 shows more details of these baselines.

4.2 Main Results

Table 2 reports the main experimental results of BCH across three datasets, from which we can observe that BCH consistently outperforms all baselines on all datasets, verifying the effectiveness of BCH. Additionally, further analysis of Table 2 leads us to the following three conclusions.

Heterogeneity is more effective than suspicious features in combating camouflaged bots. On TwiBot-20 and TwiBot-22, the two best methods both leverage heterogeneity. The best method is BCH, compared to methods based on suspicious features, it improves the F1 by 0.72% and 20.89% on these two datasets. The second-best method is BotMoE, outperforming the methods based on suspicious features by 0.40% and 0.13% on F1. These improvements highlight the importance of heterogeneity in detecting camouflaged bots.

Cross-attribute heterogeneity is more effective than intra-attribute heterogeneity in detecting nearly complete camouflaged bots. On TwiBot-22, there is a significant gap between the two best methods. As the best method, our proposed BCH surpasses the second-best BotMoE by 20.76% on F1. This indicates the notable potential of cross-attribute heterogeneity in handling nearly complete camouflaged bots.

Compared to relying on era-specific characteristics solely, incorporating suspicious features and heterogeneity facilitates BCH identifying era-diverse bots. Existing methods use either suspicious features or heterogeneity alone, hindering their ability to achieve the best for bots from different eras. For instance, some methods based on suspicious features, such as LMBot, even performer better compared to those based on heterogeneity, like BotMoE, on Cresci-15 dataset. In comparison, our BCH outperforms all methods on all datasets by

Method	$\mathcal{S}$	$\mathcal{I}$	$\mathcal{C}$	Cresci-15		TwiBot-20		TwiBot-22	
				Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score
Methods For Uncamouflaged Bots									
Wei *et al.*	✓	✗	✗	96.10 (1.1)	77.91 (0.1)	71.30 (0.2)	54.00 (2.7)	70.20 (0.1)	46.80 (1.4)
Alhosseini *et al.*	✓	✗	✗	89.60 (0.6)	87.69 (1.2)	59.90 (0.6)	57.81 (0.4)	47.72 (8.7)	29.99 (3.0)
SGBot	✓	✗	✗	77.10 (0.2)	77.91 (0.1)	81.57 (0.3)	84.90 (0.4)	75.11 (0.1)	36.59 (0.2)
BotRGCN	✓	✗	✗	96.50 (0.7)	97.30 (0.5)	83.27 (0.7)	85.26 (0.7)	77.67 (1.1)	57.50 (1.4)
Hays *et al.*	✓	✗	✗	98.00 (0.0)	97.56 (0.4)	82.10 (0.2)	86.00 (0.0)	-	-
BotPercent	✓	✗	✗	-	-	84.53 (0.3)	86.00 (0.5)	73.10 (0.0)	50.64 (0.1)
LMBot	✓	✗	✗	98.31 (0.4)	98.71 (0.1)	85.63 (0.2)	87.61 (0.3)	-	-
Methods For Initial Camouflaged Bots									
RGT	✗	✓	✗	97.20 (0.3)	97.78 (0.2)	86.60 (0.4)	87.06 (0.4)	76.50 (0.4)	42.94 (1.9)
BIC	✗	✓	✗	96.13 (0.9)	96.94 (0.2)	87.33 (0.2)	87.86 (0.2)	-	-
BotMoE	✗	✓	✗	98.00 (0.2)	98.30 (0.4)	87.30 (0.1)	88.01 (0.3)	77.81 (0.5)	57.63 (0.7)
CACL	✗	✓	✗	97.65 (2.0)	98.12 (1.3)	85.12 (1.0)	87.28 (0.8)	75.38 (0.0)	49.59 (0.8)
Methods For Nearly Complete Camouflaged and Era-Diverse Bots									
BCH (Ours)	✓	✓	✓	**98.50 (0.0)**	**98.83 (0.5)**	**87.41 (0.1)**	**88.33 (0.2)**	**78.38 (0.3)**	**78.39 (0.5)**

Table 2: Results on Cresci-15, TwiBot-20 and TwiBot-22. We run each experiment 5 times with different random seeds, and report their average results and variance. $\mathcal{S}$, $\mathcal{I}$ and $\mathcal{C}$ indicate whether suspicious features, intra- or cross-attribute heterogeneity are leveraged in corresponding method. - indicates the absence of results due to the limitations in dataset or method. **Bold** and underline indicate the best and second results. Additionally, some methods use extra posts and neighbors compared to others. To ensure fairness, we limit the number of posts/neighbors to 200/20 in reproduction, which may result in slight differences between our results and those reported in the original paper.

Settings	Average-F1	Decline
BCH (Ours)	**88.5**	**-**
(a) Effect of Different Attributes in CA Het		
1. w/o meta-attribute	86.8	1.7
2. w/o graph-attribute	86.5	2.0
3. w/o posts-attribute	86.1	2.4
(b) Effect of Different Heterogeneity		
1. w/o IA het.	85.3	3.2
2. w/o CA het.	83.9	4.6
3. w/o CA het. (topology)	82.0	6.5
(c) Effect of Joint Training Strategy		
1. $\alpha = 0$	84.8	3.7
2. $\alpha = 1$	81.2	7.3

Table 3: Ablations on the effect of different attributes in cross-attribute heterogeneity, the effect of different heterogeneity, and the effect of the joint training strategy. we report the average F1 across all three datasets.

leveraging both suspicious features and heterogeneity. This suggests that using more comprehensive characteristics helps enhance the compatibility of detection methods with both past and future.

4.3 Ablation Study

Effect of Different Attributes in Cross-attribute Heterogeneity As shown in Table 3(a), we conduct ablations by gradually removing different attributes from the cross-attribute sequence. In reality, due to the varying costs of camouflage, graph-attribute is the least camouflaged, as disguising it requires entire accounts. In contrast, the posts-attribute is the most camouflaged, as it only needs to steal posts. As for meta-attribute, its level of camouflage falls in between. Therefore, removing either of the first two attributes, compared to meta-attribute, will more severely weaken the overall cross-attribute heterogeneity, thereby causing greater impacts on detection. The ablation results clearly reflect this real-world scenario. Compared to meta-attribute, removing graph-attribute and posts-attribute result in more significant performance degradation, showing the reliability of cross-attribute heterogeneity we capture.

Effect of Different Heterogeneity As shown in Table 3(b), we conduct ablations by removing different branches from the heterogeneous encoder. The results show that removing any branch leads to performance decline, indicating that each heterogeneity contributes to detection. Moreover, the performance drop from removing cross-attribute heterogeneity far exceeds removing intra-attribute heterogeneity, highlighting the importance of cross-attribute heterogeneity; and the performance loss caused by removing topology-aware heterogeneity is greater than regular heterogeneity, emphasizing the necessity of adopting graph-based models.

Effect of Joint Training Strategy As shown in Table 3(c), we conduct ablations by fixing α in

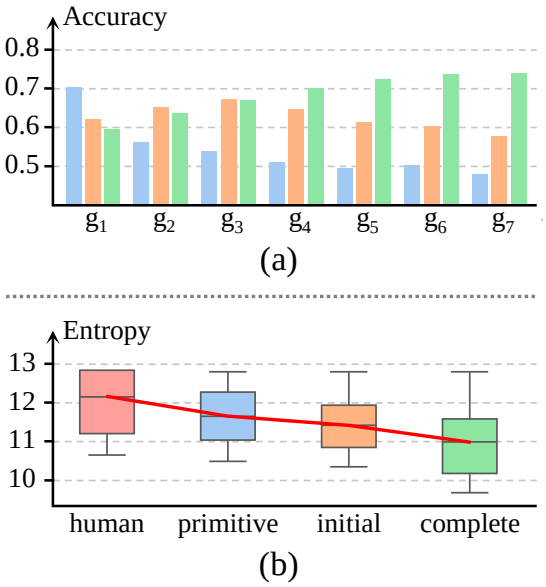

Figure 3: (a) Blue, orange and green respectively represent detection with only suspicious features, intra-attribute heterogeneity and cross-attribute heterogeneity. g_1 to g_7 represent bots from early to nearest. (b) The red line indicates the downward trend of entropy.

E.q (8) to 0 and 1. Fixing α means that BCH will no longer consider which era the bots belong to. When $\alpha = 0$, BCH treats all bots as camouflaged, while $\alpha = 1$, BCH treats all bots as uncamouflaged. Both configurations result in performance drop, suggesting that suspicious features and heterogeneity cannot fully replace each other, thus highlighting the importance of joint training.

5 Analysis

5.1 Study on Ever-Evolving Bots Detection

Cross-Attribute Heterogeneity Is an Evolutionary Invariant From an overall timeline perspective, recent bots are more advanced than earlier ones. Therefore, we can study the changes of different characteristics during the evolution of bots by grouping the bots through their account creation time. Based on this idea, we first divide the bots in the test set of TwiBot-22 into seven groups. Next, by using the training set of TwiBot-22, we train three different versions of BCH, including: detection only uses suspicious features, detection only uses intra-attribute heterogeneity, and detection only uses cross-attribute heterogeneity. Finally, we analyze the performance of these three versions on different groups. Figure 3(a) presents the results, from which we can draw two conclusions. (i) With the evolve of bots, relying solely on suspicious features or inta-attribute heterogeneity leads to performance drops, indicating they are not evolution invariants and cannot be used to detect ever-evolving bots. (ii) In contrast, detection using only cross-attribute heterogeneity maintains consistently outstanding performance, suggesting it is an evolution invariant and thus can be used to effectively handle ever-evolving bots. Furthermore, we observe that for earlier bots, performance with heterogeneity is worse than suspicious features, which also shows the necessity of joint training.

Attention Weights Can Capture Cross-attribute Heterogeneity We want to analyze whether the attention weights in BCH can reliability capture cross-attribute heterogeneity. To this end, we first use the bot type identification method mentioned in §4.1 to classify the bots in TwiBot-22 into three groups, including: uncamouflaged, initially camouflaged, and nearly complete camouflaged bots. Next, when testing the BCH trained on TwiBot-22, we visualize the cross-attribute heterogeneity weight h_c of BCH by calculating its entropy. Figure 3(b) presents the results, from which we can observe that h_c corresponding to bots has a lower entropy, differing from humans. This indicates that if h_c is used to represent cross-attribute heterogeneity, the cross-attribute heterogeneity of bots will be more stable than humans, and the stability will increase as the degree of camouflage approaches completeness. This consistency with reality demonstrates the effectiveness of attention weights in capturing cross-attribute heterogeneity.

5.2 Study on Era-Diverse Bots Detection

Method	Train \ Test	CD	PD	Average-F1
BCH	CD	**73.4**	**75.4**	**74.4**
BotMoE	CD	64.3	70.2	67.3
LMBot	CD	62.0	44.9	53.5
BCH	PD	**46.1**	**82.9**	**64.5**
BotMoE	PD	46.0	81.6	63.8
LMBot	PD	45.6	82.5	64.1

Table 4: The cross-dataset validation experiment.

To demonstrate that BCH is more compatible, we first construct two datasets using TwiBot-22, named CD (Camouflaged Dataset) and PD (Primitive Dataset). Both datasets contains same human samples but differ in bots, where CD includes only camouflaged bots and PD consists only uncamouflaged bots. Next, we select LMBot and BotMoE as

Method	Accuracy	F1-Score
BotSay (Feng et al., 2024)	89.9	91.5
BotSay (Llama2-7B)	83.5	83.5
BotSay + CA Het (Llama2-7B)	86.6	86.2

Table 5: Bots Detection with LLMs. The original BotSay use GPT-3.5-Turbo as its backbone. To reduce costs, we reproduce it on Llama2-7B. As an exploratory experiment, we select small-scale TwiBot-20 as our dataset.

our baselines, where the former relies exclusively on suspicious features and the latter solely on heterogeneity. Afterward, we split CD and PD into train/test subsets, and perform cross-dataset validation for BCH, LMBot and BotMoE using different subsets. Table 4 shows the experimental results, from which we can observe that BCH achieves the best performance on the unselected bots type, regardless of whether it is trained on camouflaged or uncamouflaged bots, indicating that BCH possesses the ability to be compatible with both the past and the future through its joint training strategy.

5.3 Detection with Large Language Models

Existing LLM-based bots detection methods still rely on traditional concepts of suspicious features and intra-attribute heterogeneity, which may limit the full potential of LLMs. To address this, we enable LLMs to capture cross-attribute heterogeneity through a two-stage training strategy. Specifically, in the first stage, we randomly replace one of a user's meta-attribute, graph-attribute or posts-attribute by copying attribute from others. The LLMs are then trained to identify which attribute has been replaced, making them more sensitive to heterogeneity across different attributes. As for the second stage, LLMs are trained to determine whether a user is a bot based on its three original (unreplaced) attributes. We adopt method proposed by (Feng et al., 2024) as our comparison, and use the same approach as it to convert users all attributes into text for compatibility with LLMs. Table 5 shows the results, which show that introducing cross-attribute heterogeneity significantly enhances the detection capabilities of LLMs.

6 Related Work

Social Media Bots Detection Early detection methods fall into three groups based on the attributes of users they use. The first group focuses on meta-attribute, leveraging traditional machine learning algorithms and manually designed features to model suspicious features (Lee and Kim, 2014; Beskow and Carley, 2018, 2019; Yang et al., 2020; Hays et al., 2023; Wu et al., 2023b). The second group targets posts-attribute, employing NLP techniques to model suspicious features (Wei and Nguyen, 2019; Kudugunta and Ferrara, 2018; Heidari and Jones, 2020; Luo et al., 2020; Wu et al., 2023a; Cai et al., 2024). The third group utilizes graph-attribute, representing social media as a graph and applying graph-based machine learning to model suspicious features (Feng et al., 2021c,a; Magelinski et al., 2020; Alhosseini et al., 2019; Feng et al., 2021c; Alothali et al., 2023; Tan et al., 2023). Although these methods can effectively detect primitive uncamouflaged bots by analyzing specific suspicious features, the limited consideration of heterogeneity constrains their further development, particularly in the face of feature camouflage. In contrast, our work jointly models suspicious features and heterogeneity, which significantly enhance the adaptability of our proposed method.

Heterogeneity Modeling Recent detection methods emphasize heterogeneity. They often use techniques adapted from general domain to model heterogeneity. We outline the widely used general heterogeneity modeling methods. For textual data, we can follow the work of (Bobur et al., 2020; Lei et al., 2023), capturing heterogeneity by using the attention weights. For graph data, we can capture heterogeneity using heterogeneous GNNs, as shown in (Shi et al., 2023; Fu et al., 2023; Ye et al., 2023; Feng et al., 2022a; Liu et al., 2023). Additionally, inspired by (Chen et al., 2024), we can also leverage contrastive learning to capture heterogeneity. However, unlike our work, existing detection methods use these general techniques to model only intra-attribute heterogeneity, while neglecting cross-attribute heterogeneity. This limit their ability to handle ever-evolving bots.

7 Conclusion

We point out that overlooking evolutionary invariant and incompatibility with the past are two weaknesses of existing works, hindering the detection of ever-evolving and era-diverse bots. To address these, we propose cross-attribute heterogeneity as invariant, and combine it with early-era bots characteristics for detection. Experiments and analysis show the effectiveness and rationale of our work in handling ever-evolving and era-diverse bots.

Limitations

In this paper, we propose only one type of evolutionary invariant, which we refer to as cross-attribute heterogeneity. In fact, we can draw on the idea used in designing cross-attribute heterogeneity to mine more invariants by introducing multimodal information. For example, if we only consider the text modality of the posts-attribute, the bot may not exhibit any obvious characteristics. However, if we also consider the images attached to the posts, cross-modal heterogeneity still exists among the bot. This heterogeneity is determined by the bot's purpose. Specifically, bots cannot fully replicate all the modal information from human posts. If they do so, they will be unable to carry out their intended malicious behavior via the posts-attribute. Instead, they only steal the text modality from human posts and hide their true intent in other modalities, such as images or videos. Therefore, cross-modal heterogeneity can also be considered as an invariant. Capturing cross-modal heterogeneity requires additional multimodal techniques, which is why we have not delved deeper into this aspect. We hope future work will address this limitation.

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 of The 2019 World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 148–153. ACM.

Eiman Alothali, Kadhim Hayawi, and Hany Alashwal. 2023. Sebd: A stream evolving bot detection framework with application of pac learning approach to maintain accuracy and confidence levels. *Applied Sciences*, 13(7):4443.

David M Beskow and Kathleen M Carley. 2018. Bothunter: a tiered approach to detecting & characterizing automated activity on twitter. In *Conference paper. SBP-BRiMS: International conference on social computing, behavioral-cultural modeling and prediction and behavior representation in modeling and simulation*, volume 3.

David M. Beskow and Kathleen M. Carley. 2019. Its all in a name: detecting and labeling bots by their name. *Comput. Math. Organ. Theory*, 25(1):24–35.

Mukhsimbayev Bobur, Kuralbayev Aibek, Bekbaganbetov Abay, and Fuad Hajiyev. 2020. Anomaly detection between judicial text-based documents. In *2020 IEEE 14th International Conference on Application of Information and Communication Technologies (AICT)*, pages 1–5. IEEE.

Zijian Cai, Zhaoxuan Tan, Zhenyu Lei, Zifeng Zhu, Hongrui Wang, Qinghua Zheng, and Minnan Luo. 2024. Lmbot: Distilling graph knowledge into language model for graph-less deployment in twitter bot detection. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining, WSDM 2024, Merida, Mexico, March 4-8, 2024*, pages 57–66. ACM.

Sirry Chen, Shuo Feng, Songsong Liang, Chen-Chen Zong, Jing Li, and Piji Li. 2024. CACL: community-aware heterogeneous graph contrastive learning for social media bot detection. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 10349–10360. Association for Computational Linguistics.

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

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

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 on World Wide Web, WWW 2016, Montreal, Canada, April 11-15, 2016, Companion Volume*, pages 273–274. ACM.

Ashok Deb, Luca Luceri, Adam Badawy, and Emilio Ferrara. 2019. Perils and challenges of social media and election manipulation analysis: The 2018 US midterms. In *Companion of The 2019 World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 237–247. ACM.

Shangbin Feng, Zhaoxuan Tan, Rui Li, and Minnan Luo. 2022a. Heterogeneity-aware twitter bot detection with relational graph transformers. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 3977–3985. AAAI Press.

Shangbin Feng, Zhaoxuan Tan, Herun Wan, Ningnan Wang, Zilong Chen, Binchi Zhang, Qinghua Zheng, Wenqian Zhang, Zhenyu Lei, Shujie Yang, Xinshun Feng, Qingyue Zhang, Hongrui Wang, Yuhan Liu, Yuyang Bai, Heng Wang, Zijian Cai, Yanbo Wang, Lijing Zheng, Zihan Ma, Jundong Li, and Minnan Luo. 2022b. Twibot-22: Towards graph-based twitter bot detection. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Shangbin Feng, Herun Wan, Ningnan Wang, Jundong Li, and Minnan Luo. 2021a. SATAR: A self-supervised approach to twitter account representation learning and its application in bot detection. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 3808–3817. ACM.

Shangbin Feng, Herun Wan, Ningnan Wang, Jundong Li, and Minnan Luo. 2021b. Twibot-20: A comprehensive twitter bot detection benchmark. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 4485–4494. ACM.

Shangbin Feng, Herun Wan, Ningnan Wang, and Minnan Luo. 2021c. Botrgcn: Twitter bot detection with relational graph convolutional networks. In *ASONAM '21: International Conference on Advances in Social Networks Analysis and Mining, Virtual Event, The Netherlands, November 8 - 11, 2021*, pages 236–239. ACM.

Shangbin Feng, Herun Wan, Ningnan Wang, Zhaoxuan Tan, Minnan Luo, and Yulia Tsvetkov. 2024. What does the bot say? opportunities and risks of large language models in social media bot detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 3580–3601. Association for Computational Linguistics.

Chengqi Fu, Shuhao Shi, Yuxin Zhang, Yongmao Zhang, Jian Chen, Bin Yan, and Kai Qiao. 2023. Squeezegcn: Adaptive neighborhood aggregation with squeeze module for twitter bot detection based on gcn. *Electronics*, 13(1):56.

Chris Hays, Zachary Schutzman, Manish Raghavan, Erin Walk, and Philipp Zimmer. 2023. Simplistic collection and labeling practices limit the utility of benchmark datasets for twitter bot detection. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pages 3660–3669. ACM.

Maryam Heidari and James H. Jones. 2020. Using BERT to extract topic-independent sentiment features for social media bot detection. In *11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference, UEMCON 2020, New York City, NY, USA, October 28-31, 2020*, pages 542–547. IEEE.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

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

Sangho Lee and Jong Kim. 2014. Early filtering of ephemeral malicious accounts on twitter. *Comput. Commun.*, 54:48–57.

Zhenyu Lei, Herun Wan, Wenqian Zhang, Shangbin Feng, Zilong Chen, Jundong Li, Qinghua Zheng, and Minnan Luo. 2023. BIC: twitter bot detection with text-graph interaction and semantic consistency. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10326–10340. Association for Computational Linguistics.

Shilong Li, Boyu Qiao, Kun Li, Qianqian Lu, Meng Lin, and Wei Zhou. 2023. Multi-modal social bot detection: Learning homophilic and heterophilic connections adaptively. In *Proceedings of the 31st ACM International Conference on Multimedia, MM 2023, Ottawa, ON, Canada, 29 October 2023- 3 November 2023*, pages 3908–3916. ACM.

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

Yuhan Liu, Zhaoxuan Tan, Heng Wang, Shangbin Feng, Qinghua Zheng, and Minnan Luo. 2023. Botmoe: Twitter bot detection with community-aware mixtures of modal-specific experts. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 485–495. ACM.

Linhao Luo, Xiaofeng Zhang, Xiaofei Yang, and Weihuang Yang. 2020. Deepbot: a deep neural network based approach for detecting twitter bots. In *IOP Conference Series: Materials Science and Engineering*, volume 719, page 012063. IOP Publishing.

Thomas Magelinski, David M. Beskow, and Kathleen M. Carley. 2020. Graph-hist: Graph classification from latent feature histograms with application to bot detection. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 5134–5141. AAAI Press.

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 *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*, pages 593–607. Springer.

Shuhao Shi, Kai Qiao, Zhengyan Wang, Jie Yang, Baojie Song, Jian Chen, and Bin Yan. 2023. Muti-scale graph neural network with signed-attention for social bot detection: A frequency perspective. *CoRR*, abs/2307.01968.

Zhaoxuan Tan, Shangbin Feng, Melanie Sclar, Herun Wan, Minnan Luo, Yejin Choi, and Yulia Tsvetkov. 2023. Botpercent: Estimating bot populations in twitter communities. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 14295–14312. Association for Computational Linguistics.

Feng Wei and Uyen Trang Nguyen. 2019. Twitter bot detection using bidirectional long short-term memory neural networks and word embeddings. In *First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications, TPS-ISA 2019, Los Angeles, CA, USA, December 12-14, 2019*, pages 101–109. IEEE.

Jun Wu, Xuesong Ye, and Yanyuet Man. 2023a. Botrinet: A unified and efficient embedding for social bots detection via metric learning. In *11th International Symposium on Digital Forensics and Security, ISDFS 2023, Chattanooga, TN, USA, May 11-12, 2023*, pages 1–6. IEEE.

Jun Wu, Xuesong Ye, and Chengjie Mou. 2023b. Botshape: A novel social bots detection approach via behavioral patterns. *CoRR*, abs/2303.10214.

Kai-Cheng Yang, Onur Varol, Pik-Mai Hui, and Filippo Menczer. 2020. Scalable and generalizable social bot detection through data selection. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 1096–1103. AAAI Press.

Sen Ye, Zhaoxuan Tan, Zhenyu Lei, Ruijie He, Hongrui Wang, Qinghua Zheng, and Minnan Luo. 2023. HOFA: twitter bot detection with homophily-oriented augmentation and frequency adaptive attention. *CoRR*, abs/2306.12870.

A Appendix

A.1 The Details of Datasets

Statistics We adopt three widely used datasets for our experiments, including Cresci-15 (Cresci et al., 2015), TwiBot-20 (Feng et al., 2021b) and TwiBot-22 (Feng et al., 2022b). Table 6 presents the statistical information of these datasets. We follow the official setup to divide these datasets into training, validation and test sets.

Statistics	C-15	T-20	T-22
# Human	1,950	5,237	860,057
# Bot	3,351	6,589	139,943
# User	5,301	229,580	1,000,000
# Post	2,827,757	33,488,192	88,217,457
# Edge	7,085,134	33,716,171	170,185,937

Table 6: The statistics of the datasets.

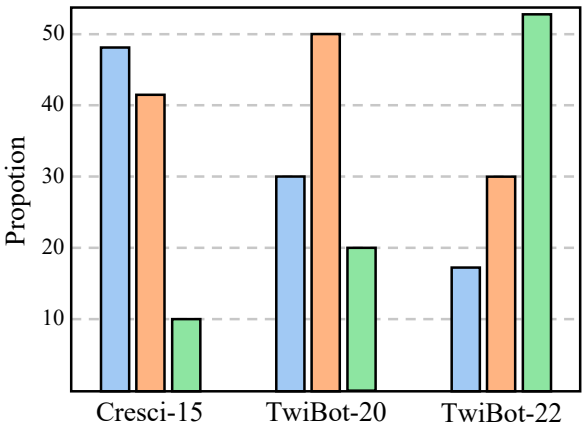

Figure 4: The proportion of different bots in each dataset. Blue, orange and green respectively represent uncamouflaged bots, initial camouflaged bots and nearly complete camouflaged bots.

LLM-Based Bot Type Identification Method To study the proportions of nearly complete camouflaged, initial camouflaged and uncamoufalged bots in each dataset, we propose a LLM-based bot type identification method, which evaluates the bot's level of camouflage based on its posts-attribute. Specifically, we first input the posts-attribute of each bot into two different LLMs, including *gpt-3.5-turbo* and *claude-3-haiku*. Next, we utilize standard zero-shot CoT (Kojima et al., 2022) to prompt each LLM to determine whether the posts-attribute belongs to a bot. Finally, we classify each bot through the judgment results generated by different LLMs. If both LLMs make the correct judgments, we consider the bot to be a uncamouflaged bot. And if one of LLMs makes an incorrect judgment, we consider the bot to be an initial camouflaged bot. Alternatively, if both LLMs make incorrect judgments, we consider the bot to be a nearly complete camouflaged bot. Figure 4 shows the results, from which we can observe that Cresci-15 is dominated by uncamouflaged bots, TwiBot-20 is dominated by initial camouflaged bots and TwiBot-22 is dominated by nearly complete camouflaged bots.

Custom Dataset In §5.2, we construct two new datasets which we refer to as CD and PD. CD includes 5,000 humans and 5,000 camouflaged bots, and PD consists of 5,000 humans and 5,000 uncamouflaged bots. Additionally, we split these datasets into training and test sets in a ratio of 8:2.

A.2 The Details of Baselines

- **CACL** (Chen et al., 2024) captures heterogeneity on graph-attribute through a contrastive learning framework for social media bots detection.
- **LMBot** (Cai et al., 2024) models suspicious features on posts-attribute by pre-training a lanaguage model for social media bots detection.
- **BotMoE** (Liu et al., 2023) simultaneously captures heterogeneity on graph-attribute and twees-attribute through a universal framework for social media bots detection.
- **BIC** (Lei et al., 2023) captures heterogeneity on twees-attribute through the attention mechanism for social media bots detection.
- **BotPercent** (Tan et al., 2023) models suspicious features on posts-attribute and graph-attribute by integrating multiple advanced methods for social media bots detection.
- **Hays *et al*** (Hays et al., 2023) models suspicious features on meta-attribute and posts-attribute through a decision tree method for social media bots detection.
- **RGT** (Feng et al., 2022a) captures graph-attribute heterogeneity by leveraging a relational graph transformer and a semantic attention network for social media bots detection.
- **BotRGCN** (Feng et al., 2021c) models suspicious features on meta-attribute, posts-attribute and graph-attribute by leveraging pre-trained language model and graph convolutional neural networks for social media bots detection.
- **SGBot** (Yang et al., 2020) models suspicious feature on meta-attribute and feeds them into random forest classifiers for social media bots detection.
- **Alhosseini *et al*** (Alhosseini et al., 2019) models suspicious feature on graph-attribute by leveraging graph convolutional neural networks for social media bots detection.
- **Wei *et al*** (Wei and Nguyen, 2019) models suspicious feature on posts-attribute by leveraging word-embeddings and bidirectional LSTMs for social media bots detection.

A.3 The Details of Training

Hyperparameter	C-15	T-20	T-22
learning rate	1e-4	1e-4	1e-4
batch size	64	64	64
epoch	30	15	10
L2 regularization	1e-5	1e-5	1e-5
Optimizer	RAdamW	RAdamW	RAdamW
Dropout	0.1	0.1	0.1

Table 7: Hyperparameters of training on each dataset.

Training Hyperparameters Table 7 presents the hyperparameters settings for training on Cresci-15, TwiBot-20 and TwiBot-22. Except for epoch, the other hyperparameters remain consistent across all the three datasets.

Training Overhead Our training is conducted on an NVIDIA GeForce RTX 3090 GPU with 24GB of memory. Training for one epoch on Cresci-15 takes approximately 0.1 GPU hours, on TwiBot-20 requires about 0.5 GPU hours, and on TwiBot-22 takes around 10 GPU hours. Our inference is performed on the same device. Inference for one epoch on Cresci-15 takes approximately 0.06 GPU hours, on TwiBot-20 requires about 0.1 GPU hours, and on TwiBot-22 takes around 0.8 GPU hours.

A.4 The Details of Model Architecture

Table 8 presents the details of BCH's architecture. Additionally, Table 9 further illustrates how the hyperparameters in the architecture are determined.

Encoder	Sub-component	Atomic-component	Input Shape	Output Shape	Purppose
Suspicious Features Encoder	MLP_Block$_{bool}$	Normalization Layer	$R^{BS\times\|B\|}$	$R^{BS\times\|B\|}$	Normalize boolean information and perform embedding
		Fully Connected Layer	$R^{BS\times\|B\|}$	$R^{BS\times4\|B\|}$	
		Fully Connected Layer	$R^{BS\times4\|B\|}$	$R^{BS\times\frac{D}{3}}$	
	MLP_Block$_{num}$	Normalization Layer	$R^{BS\times\|V\|}$	$R^{BS\times\|V\|}$	Normalize numerical information and perform embedding
		Fully Connected Layer	$R^{BS\times\|V\|}$	$R^{BS\times4\|V\|}$	
		Fully Connected Layer	$R^{BS\times4\|V\|}$	$R^{BS\times\frac{D}{3}}$	
	RoBERTa	RoBERTa-base	-	$R^{BS\times D}$	Embed textual Information
		Fully Connected Layer	$R^{BS\times D}$	$R^{BS\times4D}$	
		Fully Connected Layer	$R^{BS\times4D}$	$R^{BS\times\frac{D}{3}}$	
-	RoBERTa	RoBERTa-base	-	$R^{BS\times D}$	Embed posts-attribute
		Fully Connected Layer	$R^{BS\times D}$	$R^{BS\times4D}$	
		Fully Connected Layer	$R^{BS\times4D}$	$R^{BS\times D}$	
Heterogeneity Encoder	Transformer-Encoder	Head_4_Transformer-Encoder	$R^{BS\times seq\times D}$	$R^{BS\times seq\times D}$	Embed heterogeneity
	CNN_Block	Kernel_3_Padding_1_Stride_1_Convolutional Layer	$R^{BS\times HEAD\times H\times W}$	$R^{BS\times M_1\times H\times W}$	Downsample the heterogeneity attention weights
		Kernel_3_Stride_1_Maxpooling Layer	$R^{BS\times M_1\times H\times W}$	$R^{BS\times M_1\times 73\times 73}$	
		Kernel_3_Padding_1_Stride_1_Convolutional Layer	$R^{BS\times M_1\times 73\times 73}$	$R^{BS\times M_2\times 73\times 73}$	
		Kernel_3_Stride_1_Maxpooling Layer	$R^{BS\times M_2\times 73\times 73}$	$R^{BS\times M_2\times 24\times 24}$	
		Kernel_3_Padding_1_Stride_1_Convolutional Layer	$R^{BS\times M_2\times 24\times 24}$	$R^{BS\times M_2\times 24\times 24}$	
		Kernel_3_Stride_1_Maxpooling Layer	$R^{BS\times M_2\times 24\times 24}$	$R^{BS\times M_2\times 8\times 8}$	
		Fully Connected Layer	$R^{BS\times 64M_2}$	$R^{BS\times 256M_2}$	
		Fully Connected Layer	$R^{BS\times 256M_2}$	$R^{BS\times\frac{D}{3}}$	
	R-GCN_Block	Relation_4 R-GCN Layers	$R^{BS\times(\|fr\|+\|fw\|)\times D}$	$R^{BS\times(\|fr\|+\|fw\|)\times D}$	Embed topology-aware cross-attribtue heterogeneity
		Relation_4 R-GCN Layers	$R^{BS\times(\|fr\|+\|fw\|)\times D}$	$R^{BS\times(\|fr\|+\|fw\|)\times D}$	
		Fully Connected Layer	$R^{BS\times D}$	$R^{BS\times4D}$	
		Fully Connected Layer	$R^{BS\times4D}$	$R^{BS\times\frac{D}{3}}$	

Table 8: The details of BCH's Architecture. The light-green background highlights the first occurrence of each hyperparameter.

Name	Description	Value
BS	Batch size, which we have already discussed in §A.3	64
|B|	Boolean information, which we will introduce in Table 10	11
D	Default output dimension of RoBERTa-base	768
|V|	Numerical information, which we will introduce in Table 10	5
HEAD	Number of attention heads in the Transformer-Encoder	4
H	Corresponding to 200 posts, 10 followers and 10 followings	221
W	Corresponding to 200 posts, 10 followers and 10 followings	221
M_1	The number of output channels for the first convolutional kernel	32
M_2	The number of output channels for the second convolutional kernel	64
|fr|	Number of followers we leveraged in BCH	10
|fw|	Number of followings we leveraged in BCH	10

Table 9: Hyperparameters of the BCH's architecture.

Information Name	Description
Boolean Information	
protected	Protected or not
verified	Verified or not
geo_enabled	Enable geo-location or not
contributors_enabled	Enable contributors or not
is_translator	Tanslator or not
is_translation_enabled	Translation or not
profile_background_tile	The background tile
profile_user_background_image	Have background image or not
has_extended_profile	Have extended profile or not
default_profile	The default profile
default_profile_image	The default profile image
Numerical Information	
# follower	Number of followers
# following	Number of following
# favorites	Number of likes
# statuses	Number of statuses
# active_days	Account creation duration

Table 10: Boolean and numerical information we used in the suspicious encoder.