GMAE2: Stacking Graph Masked Autoencoder on Feature Autoencoder for Social Bot Detection

Haitao Huang[1,2*] and Mohan Zhao[3]

[1] National Key Laboratory for Multi-modal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 101408, China
3 Beijing 101 Middle School, Beijing 100091, China
huanghaitao2021@ia.ac.cn

Abstract. Currently, due to the significant negative impact of social bots, there has been widespread interest among researchers in automating the detection of social bots. And Graph Neural Network-based (GNN-based) detection methods have flourished, showing a very promising prospect in terms of detection performance. However, existing GNN-based social bot detection methods generally rely on densely annotated nodes in the context of social bot detection, leveraging them as training samples to guide the model training process, i.e., the detection social bot detection process. This demand for a large number of annotated nodes severely restricts the availability of GNN-based methods. To address this issue, we construct a GNN-based method that operates in a self-supervised pretraining-probing manner by stacking a **g**raph **m**asked **a**uto**e**ncoder on top of a feature **a**uto**e**ncoder (GMAE2). Benefiting from the pre-training of the encoder with self-supervised learning, the requirement of labeled nodes is significantly reduced. Through extensive experiments, we showcased that our GMAE2 is more suitable for social bot detection with an extremely low proportion of labeled nodes compared to existing methods. Our code is available at: https://github.com/CASI-Ahht/GMAE2-SBD.

Keywords: Social Bot Detection, Graph Self-supervised Learning, Graph Masked Autoencoder.

1 Introduction

Social bots refer to accounts that operate in an automated or semi-automated manner on social media platforms, comprising approximately 9% to 15% of the monthly active user numbers on mainstream social media [1, 2]. Apart from stealing personal privacy and causing economic losses [3, 4], social bots can also influence public opinion and threaten the integrity of democratic politics and social stability [5]. They have facilitated the spread of anti-science discourse about epidemic prevention [6], and potentially

*Corresponding author.

swaying the fairness of political elections across various regions [7, 8]. Given the vast number of social media users and social bots, the automated detection of social bots has received extensive attention from researchers.

There are three types of social bot detection methods according to their main information sources: feature-based, structure-based and hybrid methods [9, 10]. And GNN-based methods are a branch of hybrid methods, which utilize GNN encoders to integrate user feature information and structural information in an end-to-end manner for the detection of social bots. This end-to-end manner enables that GNN-based methods not only inherit the superior performance and robustness of hybrid methods but also represents a more promising prospect for future developments.

Even though GNN-based methods have their advantages in performance, most existing GNN-based methods rely on the intense labeled nodes as training samples in the context of social bot detection. However, there has been no research on reducing the demand for labeled sample in GNN-based detection methods. Until recently, Zhou et. al. [11] conduct the first exploration into leveraging contrastive learning to pre-train GNN encoders for the task of social bot detection. Nevertheless, their approach is still constrained by the instance discrimination pretext task and the structural invariance assumption, leading to its poor detection performance. Therefore, we propose a social bot detection method that leverages pretraining and probing paradigm, employing feature autoencoders and graph masking autoencoders to construct the model. In specific, our approach stacks the graph masked autoencoder on top of the feature autoencoder, with each trained in sequence. And this strategy has been shown to significantly improve training stability and detection performance. After the pretraining, a newly-initialized 2-layer multi-layer perceptron (MLP) classifier is stacked on the GNN encoder, which is trained by a small amount of labeled nodes, to detect social bots. The main contributions of this work can be summarized as follows:

1) We propose a social bot detection method via stacking **g**raph **m**asked **a**uto**e**ncoder on top of feature **a**uto**e**ncoder (GMAE2), which detect social bots in a self-supervised pretraining-probing manner, resulting in its limited labeled nodes demand.

2) Extensive experiments demonstrate that the proposed GMAE2 achieves SOTA performance on two social bot detection benchmarks under very low training set ratio settings. And ablation studies further validate each strategy's efficacy in our model.

2 Related Work

2.1 GNN-based Social Bot Detection

GNN-based social bot detection algorithms exploit GNN to jointly extract users' features and structure information, to generate users' representations. Alhosseini et al. [12] were the first to apply GNN for social bot detection. Subsequently, Feng et al. utilized relational graph convolutional network (RGCN) [13] and relational Graph Transformer (RGT) [14] to detect social bots, markedly improving detection performance. Based on RGCN, they also integrated a cross-modal attention mechanism and a mixture of expert models into the detection method [15]. Additionally, Peng et al. [16] proposed to a cross-platform detection method based on federated learning. Meanwhile, other researchers

have sought to employ graphical models to rectify the classification errors generated by GNNs. For example, Sun et al. [17] incorporated node credibility estimation through short random walks to weigh the message passing process of graph convolutional network (GCN). Deng et al. [18, 19] utilized the Markov Random Field (MRF) to rectify the errors generated by GCN, whose parameters are trained via mean field approximation and expectation-maximization (EM) algorithm. Furthermore, recently, some researchers recognized the necessity of modeling both homophily and heterophily in the context of social robot detection, and have conducted some relevant researches [20-22].

Although GNN-based social bot detection methods have flourished, these methods generally rely on large numbers of annotated samples in the detection environment to improve their performance, resulting in a lack of research in this field on methods to reduce the requirement for annotated samples. Until recently, Zhou et al. [11] conducted related research through pretraining and finetuning a GNN-based model to detect social bots. However, their method is constrained by contrastive learning and graph structure augmentation, which fails to prioritize the extraction of user feature information, resulting in its unsatisfactory performance.

2.2 Graph Self-supervised Learning

Before the popularity of contrastive learning, graph self-supervised learning methods were typically designed heuristically, such as VGAE [23], GraphRNN[24], GPT-GNN [25]. These methods typically entail performing a pretext task of reconstructing the neighbor structure, which leads to the learnt node representations being effectively tailored only for link prediction. Subsequently, with the development of contrastive learning, it has also become a mainstream approach in graph self-supervised learning. A series of works such as DGI [26], GraphCL [27], GCA [28], have emerged. These works commonly employ negative samples to prevent collapse and use data augmentation to construct positive samples to ensure that the model learns specific invariances. Afterwards, in order to address the requirement of massive negative samples in contrastive learning methods, contrastive learning approaches without negative samples have been proposed, such as CCA-SSG [29] and BGRL [30]. However, they still can't overcome the issue of reliance on specific invariances assumption. It wasn't until the introduction of GraphMAE [31] that a new method was provided for graph self-supervised learning and for resolution of reliance on invariances assumptions.

Due to the training processes of GraphMAE, it excels in extracting feature information and relationships between node features, what is precisely needed for social bot detection. Moreover, it is not bound by specific invariances assumptions, which allows us to overlook the invariance associations in the contexts of social bot detection. Therefore, we employ GraphMAE to train our GNN encoder in this paper.

3 Methods

3.1 Overview of our GMAE2

Our GMAE2 comprises a feature autoencoder, a graph masked autoencoder and a 2-layer MLP classifier. In the pretraining phase, the feature autoencoder, whose function is to merge features of different categories, is trained separately at first. Subsequently, the graph masked autoencoder, whose function is depicting associations between adjacent nodes, is trained separately in the representation space of feature encoder, with the feature decoder being discarded. Finally, the classifier is trained through limited labeled nodes in the probing phase, with parameters of both two encoders are frozen. The overall model architecture of GMAE2 is shown in Fig. 1.

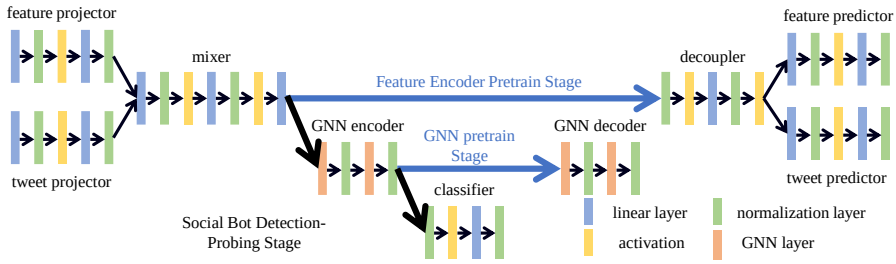

Fig. 1. The model architecture of GMAE2.

3.2 Feature Autoencoder

The feature autoencoder is utilized to extract and integrate information specific to each user. Built on the foundation of feature design, it comprises an encoder and a symmetrically structured decoder. In this autoencoder, the encoder is leveraged to blend and condense different types of information from social media users into compact representation vectors, while the decoder is used to restore the original features during pretraining phase.

Following the feature design in BotRGCN [13], we categorize the user features into four categories to incorporate users' personal information and tweet semantics: numerical, categorical, personal profile, and tweet features. Among them, numerical and categorical features are derived from users' metadata, portraying users' fundamental status. As for the personal profile features and tweet features, they represent the user's characteristics and social activities information. These features consist of the encoded user profiles and tweet semantic vectors generated by the RoBERTa model [32].

Afterwards, these features are fed into the autoencoder. At first, there are four nonlinear projection heads which are used to individually project the features of each category into fixed dimensions:

$$\hat{x}_i^j = W_2^j \varphi\left(W_1^j x_i^j + b_1^j\right) + b_2^j \tag{1}$$

where $\hat{x}_i^j$ and x_i^j represents the j-th category of dimension-reduced features and original features of the node i, the four W^j and b^j represent the learnable parameters corresponding to each category, φ represents ELU activation function, and $j \in \{1,2,3,4\}$ represents

the four distinct feature categories. Moreover, after each linear transformation, a batch normalization layer is incorporated to maintain a stable output distribution. Then, the dimension-reduced features are concatenated and merged by a three-layer MLP:

$$h_i^{(0)} = W_3\varphi(W_2\varphi(W_1 Cat\langle \hat{x}_i^1|\hat{x}_i^2|\hat{x}_i^3|\hat{x}_i^4\rangle + b_1) + b_2) + b_3 \quad (2)$$

where $Cat\langle\cdot|\cdot|\cdot|\cdot\rangle$ means concatenation operation, the six W and b are learnable parameters, and $h_i^{(0)}$ is the output of the feature encoder, which also serves as the input node features for the subsequent GNN encoder. As for the decoder, its structure is symmetrical to that of the encoder. It consists of a fusion projector followed by four category-specific feature prediction heads, with the fusion projector being solely comprised of a single linear layer.

3.3 Graph Masked Autoencoder

The graph masked autoencoder is utilized to merge users' feature and structural information into their representations. Similar to the feature autoencoder, it's also comprised of an encoder and a decoder. Furthermore, the encoder is tasked with learning clean node representations which is beneficial for the social bot detection task, while the decoder facilitates the restoration of nodes' input features from these representations. The network structure of the decoder is identical to that of the encoder, which is composed of two GNN layers proposed in our previous work. Next, a brief introduction will be provided for the GNN layer we utilize, which consists of an attention integration mechanism and two sub-GNN modules.

To capture the asymmetrical influence among users in social media context more accurately, the social network is constructed as a directed graph, where the edge directions indicate the following directions between users. And then, the directed edges in the follower direction are copied and reversed from these edges in following directions. Built on this directed graph, two sub-GNN modules are leveraged to operate in two directions simultaneously, which allows for the concurrent message propagation and information aggregation in both directions.

The role of these two sub-GNN modules, which operate on the following and follower social relationships respectively, is to recognize edge categories and perform message propagations to aggregate information which is beneficial to improve the quality of node representations. To recognize edge categories, these sub-GNN modules first generate edge representations via three linear layers, which takes nodes' representations as input:

$$\hat{h}_i^{(l),r} = \varphi\left(W_{tgt}^{(l),r} h_i^{(l-1)} + b_{tgt}^{(l),r}\right) \quad (3)$$

$$\hat{h}_j^{(l),r} = \varphi\left(W_{src}^{(l),r} h_j^{(l-1)} + b_{src}^{(l),r}\right) \quad (4)$$

$$e_{ij}^{(l),r} = W^{(l),r} Cat\langle \hat{h}_i^{(l),r}|\hat{h}_j^{(l),r}\rangle + b^{(l),r} \quad (5)$$

where $\hat{h}_i^{(l),r}$ means the dimension-reduced representations of node i in layer l under relation r, $h_i^{(l-1)}$ means the original representations of node i in layer $l-1$, $e_{ij}^{(l),r}$ means the low-dimensional representation of edge from node j to node i, and all of the W and b are learnable parameters. Subsequently, edges belonging to the same type of social relations are classified into four categories, and their message passing weights are

determined based on the inner-products between their representations and the four prototypes corresponding to the four categories:

$$y_{e_{ij}}^{(l),r} = \underset{(m,n)}{\text{argmax}}((\bar{e}_{mn}^{(l),r})^T e_{ij}^{(l),r}) \tag{6}$$

$$\alpha_{ij}^{(l),r} = \begin{cases} \sigma((\bar{e}_{y_{e_{ij}}^{(l),r}}^{(l),r})^T e_{ij}^{(l),r}) & if\ y_{e_{ij}}^{(l),r} \in \{(0,0),(1,1)\} \\ -1 \cdot \sigma((\bar{e}_{y_{e_{ij}}^{(l),r}}^{(l),r})^T e_{ij}^{(l),r}) & if\ y_{e_{ij}}^{(l),r} \in \{(0,1),(1,0)\} \end{cases} \tag{7}$$

where $y_{e_{ij}}^{(l),r} \in (m,n)$ and $\alpha_{ij}^{(l),r}$ denote the classified category and the message passing weight of edge from node j to node i, and $\bar{e}_{mn}^{(l),r}$ denotes the learnable edge embedding prototypes in layer l under relation r,where $m,n \in \{0,1\}$ and σ denotes sigmoid function.

After getting the message passing weights, an attention aggregation mechanism is used to integrate messages from two types of social relationships with the node representations from the previous layer to update node representations. At first, we utilize graph filters to generate messages from social relations according to their types. Here, we employ the normalized adjacency matrix $\hat{A}$ as the graph filter, which is normalized in both rows and columns. Through mapping the graph filter onto the spatial domain, extracting neighbor information could be expressed in the following form:

$$m_i^{(l),r} = \gamma h_i^{(l-1)} + \sum_{j \in \mathcal{N}_i^r} \frac{\alpha_{ij}^{(l),r}}{\sqrt{d_i^{in} d_j^{out}}} h_j^{(l-1)} \tag{8}$$

where $m_i^{(l),r}$ means the messages received by node i in layer l under social relation r, γ is a hyper-parameter, $\mathcal{N}_i^r$ means the neighborhood of node i, and d^{in} and d^{out} are the in-degree and out-degree of nodes respectively.

Subsequently, messages from different types of social relations are aggregated by the attention integration mechanism. Specifically, two linear layers are leveraged to merge information from three sources and calculate attention weights respectively. And then, node representations are updated through linear mixing based on the attention weights:

$$\hat{\beta}_i^{(l)} = \tanh\left(W_{int_1}^{(l)} Cat\left\langle m_i^{(l),fo} | m_i^{(l),fr} | h_i^{(l-1)} \right\rangle\right) \tag{9}$$

$$\beta_i^{(l),fo}, \beta_i^{(l),fr}, \beta_i^{(l),id} = \text{softmax}\left(\frac{W_{int_2}^{(l)} \hat{\beta}_i^{(l)}}{T}\right) \tag{10}$$

$$h_i^{(l)} = \beta_i^{(l),fo} m_i^{(l),fo} + \beta_i^{(l),fr} m_i^{(l),fr} + \beta_i^{(l),id} h_i^{(l-1)} \tag{11}$$

where $\beta_i^{(l),fo}, \beta_i^{(l),fr}$ and $\beta_i^{(l),id}$ represents integration weights for messages from follower-type relations and friend-type relations, and node's representation from the previous layer, $W_{int_1}^{(l)}$ and $W_{int_2}^{(l)}$ represent learnable parameters, and T is a hyper-parameter, with a default value of 3. Besides, specific type of social relations may be absent in the neighborhood of some nodes. To address errors caused by neighbor absence, we utilize an attention mask to exclude message inputs from corresponding type of relations during calculating integration weights.

3.4 Training in an Autoencoder Fashion

Within this research, to minimize the reliance on labeled nodes during model training, we adopt the pretraining-probing approach to cultivate a social bot detection model. In the pretraining phase, since node labels are unavailable, an autoencoder method is employed twice to train the model, aiming to retain as much information as possible. This approach is found to significantly improve the classifier's capability to accurately distinguish between the two types of users. Specifically, at first, the feature encoder is trained in an autoencoder manner. The encoder is employed to compress the original features into low-dimensional embedding vectors, which are then inputted into the decoder to reconstruct the original features. In this process, mean squared error between the reconstructed features and the initial features is utilized as the loss function to jointly train the encoder and decoder. Subsequently, the GNN encoder is trained in a graph masked autoencoder manner within the representation space of the feature encoder. In other words, rather than restoring the initial node features, the GNN decoder reconstructs the node features fed into the GNN encoder by the feature encoder merely, thus reducing the computational demands of training while mitigating interference from a less effective feature decoder. When training the GNN encoder, drawing inspiration from GraphMAE [31], a similar methodology is employed to train our GNN encoder, where the GNN decoder, the masked feature reconstruction strategy, and the cosine error loss function are pivots to its success. To summarize, as shown in Fig. 2, we initially mask a subset of node features using a learnable mask and then feed all the features into the GNN encoder. Subsequently, we apply another learnable mask to the node representations output by the GNN encoder, targeting the same subset of nodes. These masked representations are then fed into the GNN decoder, which is utilized to reconstruct the raw features of the masked nodes. Finally, we calculate the cosine error between the reconstructed and original features and back-propagate the loss to update models' parameters simultaneously.

Upon the completion of pretraining, an extra freshly initialized 2-layer MLP classifier is appended to the detection model, whose purpose is to identify whether a user is a social bot through leveraging the user representations produced by GNN encoder. It is trained in a supervised manner, employing a limited set of annotated nodes as labeled samples and employing cross-entropy as the loss function. Furthermore, the parameters of feature encoder and GNN encoder are frozen to prevent updates, which helps maintain the pretraining-probing detection paradigm and reduces the risk of overfitting associated with the limited training data.

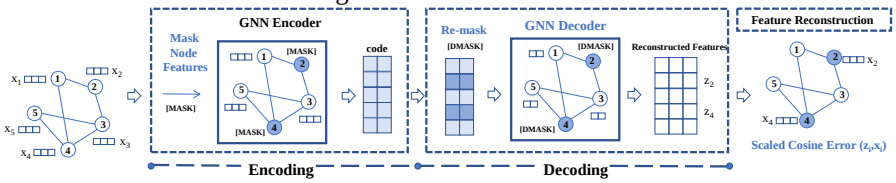

Fig. 2. The training streamline of graph masked autoencoder.

4 Experiments and Results

4.1 Experiments Setup

The performance of GMAE2 were assessed on two prominent social bot detection datasets: Twibot-20 [33], and MGTAB [34], whose statistical data are presented in Table 1. Additionally, in this paper, to better align with real-world needs in social bot detection, a 5%/15%/80% split was employed for both datasets instead of the original 70%/20%/10% split [33, 35]. Besides, due to the lack of raw data in MGTAB dataset, we employed the numerical features and tweet embedding vectors available within the dataset as the initial user features in our experiments. For detailed information on the model's architecture and hyperparameter configurations, please refer to our publicly available code repository.

Table 1. Statistical Data of Two Datasets.

Dataset	Users	Labeled Users	Edges	Ratios of Bots to Humans
Twibot-20	229580	11826	227979	1.2582
MGTAB	410199	10199	81659211	0.3688

To demonstrate the performance of our GMAE2 thoroughly, the following baselines are included in this paper:

- DAMRG [16] utilizes multiple classifiers to detect social bots in different fields for improving detection performance on the basis of learning users' representations on multi-relational graph.
- BotRGCN [13] exploits relational GCN to represent users' information and detect social bots.
- RGT [14] exploits GT to detect social bots by integrating two sets of GT designed for two types of friendships via semantic attention networks.
- SIRAN [21] utilizes GT and initial residual relation network simultaneously to address the issue of graphs with heterophily and to detect social bots.
- BothH [20] leverages feature similarity graph to improve the homophily of graph and utilizes GNN with heterophily to discern homophilic edges to improve social bot detection performance.
- CBD [11] detects social bots in a self-supervised pretraining-finetuning manner. It utilizes instance discrimination and structural data augmentation to pretrain GNN encoder in contrastive learning.

4.2 Main Results

We mainly employed two evaluation metrics, accuracy and binary F1, to assess the detection performance of all detection methods. And we employed recall as an additional metric on the MGTAB dataset, which boasts a high degree of labeling precision, thereby offering a more accurate reflection of the algorithm's performance. The main

experiment results are demonstrated in Table 2. And the following observations were made:

- Our GMAE2 achieves SOTA performance among various methods. Compared to semi-supervised baselines, GMAE2 lags slightly behind the top-performing accuracy, yet it substantially surpasses all methods in the F1 score. Considering the typically low recall rates associated with semi-supervised methods and the skewed class distribution of the MGTAB dataset, it is evident that the GMAE2 proposed by us exhibits a clear advantage in recall over these semi-supervised baselines. It suggests that our GMAE2 is more adept at identifying social bots than semi-supervised methods, leading to a more effective purification of the social media discourse environment. However, it also risks flagging more innocent human users as bots, which could increase the manual review costs.
- Compared to CBD, which detect social bots in a self-supervised pretraining and finetuning approach, our proposed GMAE2 consistently outperforms it on both datasets, with only a slight shortfall in recall. However, CBD's enhancement in recall is acquired at the expense of larger decrements in accuracy and F1, which implies that its detection outcomes exhibit a notable drop in precision when contrasted with our proposed GMAE2. And a marked decrease in precision would result in the erroneous classification of a large volume of human users as social bots, incurring considerable expenses in manual verification for both detection efforts and social media platforms, thereby substantially reducing the efficacy of the detection algorithms. Furthermore, it indicates that our GMAE2 exhibits a superior balance between recall and precision.

- **Table 2.** Main experiment results.

Methods	Twibot-20 Dataset		MGTAB Dataset		
	Acc(std)	F1(std)	Acc(std)	F1(std)	Recall(std)
DAMRG	**78.97(2.10)**	81.36(2.18)	84.72(0.42)	73.66(0.78)	79.75(1.32)
BotRGCN	77.56(0.98)	80.08(0.81)	**85.73(0.45)**	72.77(0.68)	71.17(0.98)
RGT	54.68(0.82)	41.46(2.34)	85.04(0.40)	74.95(0.51)	83.50(1.76)
SIRAN	74.05(0.63)	78.93(0.69)	85.47(0.37)	74.04(0.81)	74.41(1.85)
BothH	55.76(0.20)	71.10(1.24)	85.03(0.78)	72.19(1.48)	72.55(2.19)
CBD	77.97(0.98)	80.31(1.25)	84.24(1.12)	73.86(1.64)	**83.88(1.40)**
GMAE2	78.69(0.44)	**82.45(0.35)**	85.51(0.29)	**75.34(0.58)**	82.64(1.01)

4.3 Further Discussion

In order to further assess the performance of several methods in scenarios with extremely limited training samples, we progressively decreased the proportion of the training set by reconfiguring the training and validation sets, and chose several high-performing baselines as well as our GMAE2 for testing. The test results are presented in Fig. 2. It's evidently that our GMAE2 exhibits a relatively modest decline in performance as the training set proportion declines. when the training set percentage falls

within the 2% to 5% range, GMAE2 maintains a clear advantage over other algorithms. Nevertheless, Nevertheless, when the training set proportion is sliced down to a mere 1%, GMAE2's performance takes a drastic descent due to the overfitting encountered by the 2-layer MLP classifier, falling even below that of several other algorithms. Curiously, the RGT trained in a semi-supervised fashion exhibits no discernible drop in performance. The reasons behind its performance advantage are yet to be fully unraveled. We defer a deeper investigation into this matter to our future endeavors, where we will use the insights to refine our approach and broaden the scope of its application.

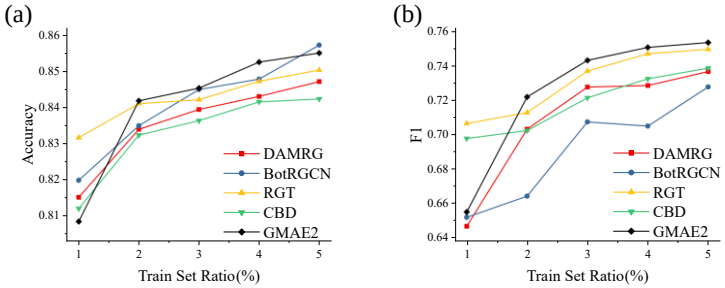

Fig. 3. Performance of Several Models Across Varying Training Set Proportions. (a) Accuracy with training set ratio changes. (b)F1 with training set ratio changes.

4.4 Ablation Study

In order to understand the contributions of different modules comprehensively, we conducted an ablation study using the MGTAB dataset. First, we investigated the efficacy of the feature encoder and the GNN encoder through discarding them respectively, namely "w\o feature encoder" and "w\o GNN encoder". Then, we investigated the efficacy of the strategy of training the feature encoder and GNN encoder separately by training them simultaneously, namely "training at the same time". Subsequently, we explored the efficacy of the masked feature reconstruction pretext task by replacing it with instance discrimination [27] or contrastive learning without negative samples [36], namely, "instance discrimination" and "no negative samples". Finally, we explored the efficacy of the pretraining-probing approach through replacing it with pretraining-finetuning approach, namely "pretraining-finetuning".

Results of ablation studies were presented in Table 3. The following observations were made:

- Feature encoder and GNN encoder are both instrumental in the detection model. Notably, the feature encoder works to minimize user information loss, which is vital for improving node separability.
- It's crucial to separate the training processes of the feature encoder and the GNN encoder to ensure the efficacy of the model's training. Training them concurrently can lead to a breakdown of the model's effectiveness, resulting in the drastic fluctuations in its performance.
- GNN encoder trained on reconstructing masked features is able to generate node representations that better preserve user feature information, which is beneficial

for enhancing the model's detection performance. Meanwhile, the poorer performance of other pretext tasks indicates that the structural invariances assumptions they encapsulate are not applicable to the social bot detection task. The node representations generated by the GNN encoder trained through these pretext tasks do not effectively preserve the difference between two types of users.

- Fine-tuning is not applicable to our proposed GMAE2, as it results in a marked decrease in model efficacy. In other words, despite achieving SOTA performance through pretraining and probing alone, we are now unable to improve the model's detection capabilities through fine-tuning. The exact reasons for this are yet to be determined, and we will delve into this issue in our upcoming research to develop a detection model with superior performance.

Table 3. Results of ablation studies.

Model	Acc	F1
w\o feature encoder	81.49	68.53
w\o GNN encoder	84.42	73.98
training at the same time	82.41	72.47
instance discrimination	84.53	74.02
no negative samples	84.73	73.78
pretraining-finetuning	74.02	65.53
GMAE2	85.51	75.34

5 Conclusion

In this paper, we proposed a GNN-based social bot detection method, GMAE2, via stacking graph masked autoencoder on the top of feature autoencoder, which detect social bots in a self-supervised pretraining-probing manner. And the feature encoder is trained on the representation space of the feature encoder, which is the main difference between our method and other methods. Subsequently, we conducted extensive experiments which demonstrated that our GMAE2 achieves SOTA performance under extremely low training set ratio settings. However, there is still room for improvement in our method: 1) When the proportion of the training set decreases to 1%, the performance of our model deteriorates significantly due to overfitting in the MLP classifier. 2) Currently, our GMAE2 is unable to use fine-tuning techniques to further improve performance. These issues remain to be explored further in the future.

Acknowledgement: This work was supported in part by the Ministry of Science and Technology of China under Grant 2020AAA0108401, in part by the National Natural Science Foundation of China under Grant 72225011 and Grant 72293575, and in part by a grant from MOE Social Science Laboratory of Digital Economic Forecasts and Policy Simulation at University of Chinese Academy of Sciences.

References

1. GRIMME C, PREUSS M, ADAM L, et al. Social bots: Human-like by means of human control? [J]. Big data, Dec. 2017, 5(4): 279-93.
2. VAROL O, FERRARA E, DAVIS C, et al. Online human-bot interactions: Detection, estimation, and characterization; proceedings of the Proceedings of the international AAAI conference on web and social media, F, 2017 [C].
3. MICHALOPOULOS D, MAVRIDIS I. Surveying Privacy Leaks Through Online Social Network; proceedings of the 2010 14th Panhellenic Conference on Informatics, F, 2010 [C]. Tripoli, Greece.
4. CRESCI S, LILLO F, REGOLI D, et al. Cashtag Piggybacking: Uncovering Spam and Bot Activity in Stock Microblogs on Twitter [J]. ACM Transactions on the Web (TWEB), Apr. 2019, 13(2): 1-27.
5. SHAO C, CIAMPAGLIA G L, VAROL O, et al. The spread of low-credibility content by social bots [J]. Nature Communications, Nov. 2018, 9(1): 1-9.
6. KAI-CHENG Y, CHRISTOPHER T-L, FILIPPO M. Prevalence of Low-Credibility Information on Twitter During the COVID-19 Outbreak [J]. arXiv preprint arXiv:200414484, 2020.
7. BESSI A, FERRARA E. Social bots distort the 2016 US Presidential election online discussion [J]. First monday, Nov. 2016, 21(11-7).
8. FERRARA E. Disinformation and Social Bot Operations in the Run Up to the 2017 French Presidential Election [J]. First monday, Aug. 2017, 22(8).
9. CRESCI S. A decade of social bot detection [J]. Communications of the ACM, Sep. 2020, 63(10): 72–83.
10. LATAH M. Detection of malicious social bots: A survey and a refined taxonomy [J]. Expert Systems with Applications, Aug. 2020, 151: 113383.
11. ZHOU M, ZHANG D, WANG Y, et al. Detecting Social Bot on the Fly using Contrastive Learning; proceedings of the Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, F, 2023 [C].
12. ALHOSSEINI S A, TAREAF R B, NAJAFI P, et al. Detect Me If You Can: Spam Bot Detection Using Inductive Representation Learning [Z]. WWW Companion. San Francisco, CA, USA. 2019: 148–53
13. FENG S, WAN H, WANG N, et al. BotRGCN: Twitter bot detection with relational graph convolutional networks; proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Netherlands, F, 2022 [C].
14. FENG S, TAN Z, LI R, et al. Heterogeneity-aware twitter bot detection with relational graph transformers; proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, F, 2022 [C].
15. LIU Y, TAN Z, WANG H, et al. BotMoE: Twitter Bot Detection with Community-Aware Mixtures of Modal-Specific Experts [J]. arXiv preprint arXiv:230406280, 2023.
16. PENG H, ZHANG Y, SUN H, et al. Domain-Aware Federated Social Bot Detection with Multi-Relational Graph Neural Networks; proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), Padova, Italy, F, 2022 [C]. IEEE.
17. SUN Y, YANG Z, DAI Y. Trustgcn: Enabling graph convolutional network for robust sybil detection in osns; proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, F, 2020 [C].
18. WU Y, LIAN D, XU Y, et al. Graph convolutional networks with markov random field reasoning for social spammer detection; proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, F, 2020 [C].

19. DENG L, WU C, LIAN D, et al. Markov-Driven Graph Convolutional Networks for Social Spammer Detection [J]. IEEE Transactions on Knowledge and Data Engineering, Feb. 2022: 12310 - 22.
20. LI S, QIAO B, LI K, et al. Multi-modal Social Bot Detection: Learning Homophilic and Heterophilic Connections Adaptively; proceedings of the Proceedings of the 31st ACM International Conference on Multimedia, F, 2023 [C].
21. ZHOU M, FENG W, ZHU Y, et al. Semi-Supervised Social Bot Detection with Initial Residual Relation Attention Networks; proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, F, 2023 [C]. Springer.
22. YE S, TAN Z, LEI Z, et al. HOFA: Twitter Bot Detection with Homophily-Oriented Augmentation and Frequency Adaptive Attention [J]. arXiv preprint arXiv:230612870, 2023.
23. KIPF T N, WELLING M. Variational graph auto-encoders [J]. arXiv preprint arXiv:161107308, 2016.
24. YOU J, YING R, REN X, et al. Graphrnn: Generating realistic graphs with deep auto-regressive models; proceedings of the International Conference on Machine Learning, F, 2018 [C]. PMLR.
25. HU Z, DONG Y, WANG K, et al. Gpt-gnn: Generative pre-training of graph neural networks; proceedings of the Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, F, 2020 [C].
26. VELIČKOVIĆ P, FEDUS W, HAMILTON W L, et al. Deep Graph Infomax; proceedings of the International Conference on Learning Representations, F, 2018 [C].
27. YOU Y, CHEN T, SUI Y, et al. Graph contrastive learning with augmentations; proceedings of the Advances in neural information processing systems, New Orleans, LA, USA, F, 2020 [C].
28. ZHU Y, XU Y, YU F, et al. Graph contrastive learning with adaptive augmentation; proceedings of the the Web Conference 2021, F, 2021 [C].
29. ZHANG H, WU Q, YAN J, et al. From canonical correlation analysis to self-supervised graph neural networks; proceedings of the Advances in neural information processing systems, Online, F, 2021 [C].
30. THAKOOR S, TALLEC C, AZAR M G, et al. Bootstrapped representation learning on graphs; proceedings of the ICLR 2021 Workshop on Geometrical and Topological Representation Learning, F, 2021 [C].
31. HOU Z, LIU X, CEN Y, et al. Graphmae: Self-supervised masked graph autoencoders; proceedings of the Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, F, 2022 [C].
32. LIU Y, OTT M, GOYAL N, et al. Roberta: A robustly optimized bert pretraining approach [J]. arXiv preprint arXiv:190711692, 2019.
33. FENG S, WAN H, WANG N, et al. TwiBot-20: A Comprehensive Twitter Bot Detection Benchmark; proceedings of the ACM International Conference on Information & Knowledge Management, Queensland, Australia, F, 2021 [C].
34. SHI S, QIAO K, CHEN J, et al. Mgtab: A multi-relational graph-based twitter account detection benchmark [J]. arXiv preprint arXiv:230101123, 2023.
35. FENG S, TAN Z, WAN H, et al. TwiBot-22: Towards graph-based Twitter bot detection; proceedings of the Advances in neural information processing systems, New Orleans, LA, USA, F, 2022 [C].
36. CHEN X, HE K. Exploring simple siamese representation learning; proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, F, 2021 [C].