Beyond Individual Mimicry: Constructing Human-Like Social network with Graph-Augmented LLM Agents

Anonymous ACL submission

Abstract

Driven by large language models (LLMs), social bot can autonomously engage in local interactions, whose human-like behaviors enable them to evade social bot detection. However, while these botnets exhibit realistic local social interactions, they fail to preserve human-like social network. This is because LLM-based bots are graph-unaware and cannot coordinate over global interactions, which makes those botnets vulnerable to graph neural network (GNN)-based detection. To address this limitation, we propose GraphMind, which equips LLM-driven social bots to explicitly learn and fit human-like social network structures. Building on this foundation, we further construct GraphMind-Botnet, a LLM-driven botnet designed to evaluate the performance of existing social bot detection algorithms. Experiments on datasets derived from GraphMind-Botnet show that both text-based and graph-based detection models show substantially degraded performance in distinguishing. Our results highlight the critical role of social link construction in LLM-driven social network generation, while exposing fundamental weaknesses in existing bot detection mechanisms.

1 Introduction

Recent advances in large language models (LLMs) have fundamentally transformed social bot research. By generating highly fluent, context-aware, and emotionally expressive content, LLM-driven social bots can closely mimic human communication behaviors, substantially narrowing the gap between automated agents and genuine users (Qiao et al., 2025; Kong et al., 2025).

Existing LLM-based social botnet simulations primarily focus on enhancing the human-likeness of individual bot behaviors. They improves single-agent realism through prompt engineering (Ekin, 2023) or parameter-efficient fine-tuning techniques (Devalal and Karthikeyan, 2018; Wu et al., 2025), enabling bots to produce fluent, emotionally expressive responses and human-like decision-making patterns.

However, while these approaches are effective at the level of individual behaviors and local interaction patterns, they often overlook a fundamental characteristic of real-world social systems: the structure of the underlying social graph. In existing LLM-driven botnets, individual agents can already be misclassified as human users by detectors that rely primarily on user metadata or textual features (Qiao et al., 2025). Nevertheless, at the network level, such botnets largely fail to establish meaningful and persistent social edges between nodes, which in practice results in a large number of isolated or weakly connected users, and therefore remain vulnerable to graph neural network (GNN)-based detection methods that exploit structural inconsistencies in the social graph (He et al., 2024; Wu et al., 2024).

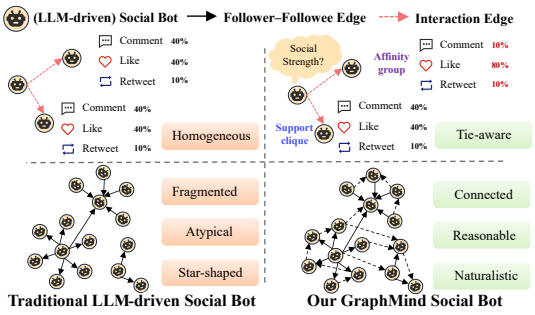


Figure 1: The comparation between existing LLM-driven social bot and GraphMind (ours). Our botnet resembles more human-like social networks, leading to significantly improved evasion performance against GNN-based detection.

This vulnerability arises from two key structural deficiencies. (1) *Existing LLM-driven botnets exhibit superficial behavioral diversity but excessive homogeneity in one-hop interactions.* In real OSNs, users modulate their interactions based on the strength and context of social ties, leading to heterogeneous engagement patterns across neighbors (Arnaboldi et al., 2016). In contrast, although LLM-driven social bots support a richer set of interaction actions, they tend to apply these actions in a largely uniform manner across neighbors. Such uniformity induces abnormal ego-network patterns that are readily captured by heterogeneous GNN-based detection methods. (2) *Current LLM-driven botnets largely lack meaningful multi-hop follow relationships.* Most LLM-driven bots operate within a small number of local communities, resulting in fragmented graphs with many isolated nodes and weak cross-community connectivity. This sharply contrasts with real social networks, which are known to be almost fully connected: Face-

book reports that 99.91% of users belong to a single giant connected component (Ugander et al., 2011).

These limitations arise from a fundamental mismatch between LLMs and social networks. LLMs are designed for sequential text modeling and are typically optimized without explicit objectives or state representations over global network topology (Bei et al., 2025; Tabassum et al., 2018). As a result, while individual interactions may appear realistic, LLM-driven botnets fail to jointly optimize graph-level properties, leading to structurally incomplete social networks.

To bridge this gap, we introduce explicit social graph knowledge into LLM-driven social bots by encoding graph information into structured, sequence-based representations compatible with LLM reasoning. We propose **GraphMind**, the first framework that integrates graph reasoning into LLM-based social behavior modeling. As shown in Figure 2 (left), GraphMind operates through two modules: (1) *Fine-Grained Interaction Modeling*, which serializes one-hop relational context (e.g., tie strength) to enable relationship-aware actions, and (2) *Graph-Augmented Social Inference*, which encodes multi-hop structural information into relational sequences, allowing bots to construct cross-community links guided by small-world principles (Milgram et al., 1967) in online social networks.

These agents collectively form the GraphMind botnet, as illustrated in Figure 2 (right). We generate the botnet through a three-stage process: first, we partition agents into communities and construct an initial intra-community network structure; second, GraphMind agents infer potential social edges (specifically, follow relationships) between previously unconnected user pairs; finally, agents generate interaction records conditioned on varying relationship strengths with their neighbors. The resulting network exhibits improved structural realism compared to prior LLM-driven botnets, substantially degrading the performance of existing GNN-based detection methods.

Our main contributions are summarized as follows:

- **GraphMind Framework.** We present the first systematic study of how LLM-based agents can learn human social network structures and proactively establish social connections during simulation, enabling the generated networks to closely match key structural properties of real-world social graphs.
- **GraphMind Dataset.** Based on GraphMind, we construct an LLM-based dataset whose bots exhibit more human-like network structures than existing LLM-based social simulation, enabling a controlled yet realistic evaluation of the robustness of social bot detection methods.
- **Evaluation and Defense Insights.** Experiments on our dataset demonstrate substantial performance degradation of state-of-the-art detectors, motivating the development of more robust social bot detection approaches.

2 Preliminaries

2.1 Problem Formulation

We model an online social network as a directed graph $G = (V, E)$, where V denotes a set of social bots and $E \subseteq V \times V$ represents directed follow relationships. Each node $v \in V$ is associated with a textual profile, encoded into a fixed-dimensional embedding $\mathbf{x}_v \in \mathbb{R}^d$ using the embedding layer of an LLM with last-token pooling. We denote the collection of node representations as $X = \{\mathbf{x}_v \mid v \in V\}$.

Beyond structural links, follow edges may carry interaction annotations such as likes, retweets, and comments, denoted as

$$Y = E_{\text{Interaction}}, \tag{1}$$

which together with E yields a heterogeneous representation of social relationships.

We assume an initial graph $G_0 = (V, E_0, Y_0)$, where E_0 is sparse and Y_0 is incomplete, resulting in fragmented structure and limited edge-level semantics.

2.2 Social Graph Completion

Given the initial graph $G_0 = (V, E_0, Y_0)$, our objective is to infer missing follow edges $\Delta E \subseteq (V \times V) \setminus E_0$ *and* their associated interaction annotations ΔY, such that the completed graph

$$G' = (V, E_0 \cup \Delta E, Y_0 \cup \Delta Y) \tag{2}$$

resembles real-world human social networks in terms of both structural and interaction-level properties.

Specifically, we aim to jointly generate ΔE and ΔY so that key graph statistics of G', including connectivity, degree heterogeneity, and short average path length, are close to those observed in real online social networks, while maintaining realistic interaction patterns along inferred social ties.

3 Methodology

As illustrated in Figure 2 (left), GraphMind comprises two complementary modules that operate at different granularities of social behavior. Fine-Grained Interaction Modeling generates relationship-aware interaction records conditioned on social tie strength, completing missing edge-level interactions ΔY. Graph-Augmented Social Inference enables LLM-driven agents to infer missing follow relationships during simulation, completing the social graph structure by generating ΔE and jointly yielding networks that are realistic in both structure and interaction dynamics.

GraphMind framework is trained using Lora fine-tuning (Shen et al.) and Group Relative Policy Optimization (GRPO) (Shao et al., 2024). In the following, we describe how to construct the dataset and design the reward function to achieve this objective.

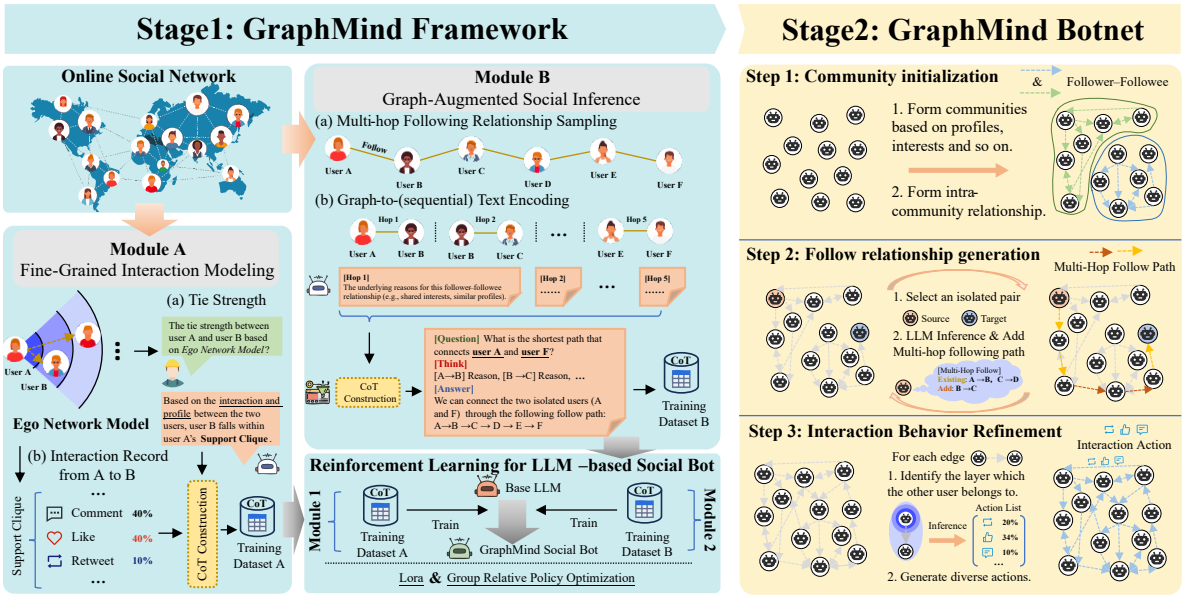


Figure 2: Overview of the framework. **(Left)** GraphMind Framework, where two modules are employed to enable LLMs to generate diverse, strength-aware interactions and to construct multi-hop follow chains, thereby mitigating isolated nodes. **(Right)** Botnet simulation, in which GraphMind social bots autonomously build human-like social networks, improving structural realism and enhancing robustness against GNN-based detection methods.

3.1 Fine-Grained Interaction Modeling

Interaction-level heterogeneity along social ties plays a critical role in shaping realistic ego-network patterns. In existing LLM-based botnets, interaction behaviors tend to be applied uniformly across neighbors, yielding one-hop interaction distributions that are overly simplistic and fail to capture the diversity of human social interactions. This module is designed to endow agents with relationship-aware interaction behaviors, thereby generating diverse and socially coherent edge-level interactions that better reflect human-like interactions.

To this end, we leverage human interaction data from TwiBot-22 (Feng et al., 2022a) as supervision. Each training instance consists of a user pair and their observed interactions, formalized as

$$Y_{\text{fine}} = \{(\mathbf{x}_u, \mathbf{x}_v, e_{\text{interaction}}, l^e) \mid (u \to v) \in E_{h\to h}\}, \tag{3}$$

where $\mathbf{x}_u$ and $\mathbf{x}_v$ denote the textual profile embeddings of the two users, l^e is the ground-truth relationship strength label defined according to the four-level ego-network hierarchy in Table 4 (Arnaboldi et al., 2016), and $e_{\text{interaction}}$ represents the sequence of observed actions along the directed edge. Based on these samples, we adopt a chain-of-thought reasoning paradigm (Wei et al., 2022) to factor interaction generation into two stages: the model first infers latent social relationship attributes from profile representations, and then generates interaction records in the MCP format (Hou et al., 2025) conditioned on the inferred relationship context. This two-stage design allows the model to disentangle relationship inference from action generation and produces interaction sequences that are both semantically and socially coherent.

We optimize the model via GRPO to jointly learn relationship inference and interaction generation. For each training sample, the model predicts a relationship level l^e_{pred} from the profile embeddings and generates a corresponding action sequence. Two complementary reward terms guide this process. The relationship inference reward

$$R_1 = \exp\big(-|l^e_{\text{pred}} - l^e|\big) \tag{4}$$

favors predictions that preserve relative social distance with respect to the ground-truth label l^e, ensuring that inferred relationships remain consistent with the observed ego-network hierarchy. To enforce action–relationship consistency at the instance level, we measure the divergence between the model-generated action distribution and the empirical distribution of the current sample,

$$R_2 = -\mathrm{KL}\big(P_\theta(\cdot \mid l^e_{\text{pred}}) \,\big\|\, P_{\text{sample}}(\cdot)\big), \tag{5}$$

where $P_\theta(\cdot \mid l^e_{\text{pred}})$ denotes the empirical distribution derived from the model's generated action sequence, and $P_{\text{sample}}(\cdot)$ denotes the empirical distribution from the ground-truth sequence $e_{\text{interaction}}$. By maximizing R_2, the model is encouraged to generate action sequences whose distribution closely follows that of the observed human interactions, conditioned on the predicted relationship level. The combined reward is

$$R_{\text{Fine}} = R_1 + R_2, \tag{6}$$

which drives the model to produce socially plausible and relationship-aware interactions while preserving the diversity and heterogeneity observed in real-world user behaviors. An example of the constructed training sample is provided in Table 7.

3.2 Graph-Augmented Social Inference

Inspired by WalkLM (Tan et al., 2023), we propose Graph-Augmented Social Inference to address a key limitation of existing LLM-based botnets: the lack of global connectivity and meaningful cross-community social ties. The core challenge lies in reconciling the non-sequential nature of graph structures with the sequential input paradigm of LLMs. To this end, GSI adopts a two-stage training strategy: (1) supervised fine-tuning on multi-hop human social chains to inject structural priors of real social networks, and (2) policy optimization to generate multi-hop path.

3.2.1 Supervised Fine-Tuning

The objective is to endow the LLM with prior knowledge of how human users form long-range social connections in real-world networks. Instead of learning from isolated edges, we construct supervision signals based on multi-hop follow paths sampled from human social graphs and serialize them into sequential reasoning traces suitable for LLM training.

Multi-hop Human Relationship Sampling. We extract multi-hop follow paths from a human-only subgraph of TwiBot-22 to serve as structural supervision. Specifically, a human user $v_{h_0} \in V_H$ is selected as the anchor node of a social chain C, which is iteratively extended along directed human-to-human follow edges. A sampled chain is defined as

$$C = [v_{h_0}, v_{h_1}, \ldots, v_{h_k}], \tag{7}$$

where each adjacent pair satisfies $e_{v_{h_i} \to v_{h_{i+1}}} \in E_{h \to h}$.

The expansion process terminates when no further outgoing human-follow edges are available or when the chain reaches a maximum length. Consistent with empirical observations of small-world social networks (Milgram et al., 1967), we restrict the maximum chain length to six hops.

To construct semantically coherent and structurally meaningful paths, node selection is guided by both semantic similarity to the anchor user and node influence. At each step, the next node is selected by

$$v_{h_{i+1}} = \arg \max_{v \in \mathcal{C}(v_{h_i})} \Big(\cos(\mathbf{x}_{v_{h_0}}, \mathbf{x}_v) + \tilde{d}^-(v) \Big), \tag{8}$$

$$\tilde{d}^-(v) = \frac{\deg^-(v)}{\max_{u \in \mathcal{C}(v_{h_i})} \deg^-(u)}, \tag{9}$$

where $\mathcal{C}(v_{h_i})$ denotes candidate human nodes reachable from v_{h_i}, $\tilde{d}^-(v)$ denotes normalized in-degree. This strategy favors paths that remain semantically grounded while naturally traversing structurally influential users, which often act as bridges across communities.

Graph-to-Sequential Text Encoding. Since LLMs do not natively operate on graph-structured inputs, each sampled multi-hop chain is transformed into a sequential natural-language representation. Given a chain $C = [v_i, \ldots, v_j]$, we decompose it into consecutive follow relations. For each hop $(v_k \to v_{k+1})$, we generate a brief textual rationale t_k explaining why user v_k follows v_{k+1}, conditioned on user profile attributes and node-level structural signals (e.g., relative in-degree). The hop-level rationales are concatenated to form a chain-of-thought (CoT) sequence as bellow

$$\text{COT}_{v_i \to v_j} = \langle \text{User} \rangle_i \oplus t_i \oplus \cdots \oplus t_{j-1} \oplus \langle \text{User} \rangle_j, \tag{10}$$

where $\oplus$ denotes textual concatenation. This representation enables the LLM to learn how long-range social connections are composed through a sequence of locally plausible follow decisions.

Loss Function. Let $\mathcal{D}_{\text{SFT}}$ denote the constructed dataset of multi-hop reasoning sequences. The supervised fine-tuning objective minimizes the negative log-likelihood of the reference CoT sequences:

$$\mathcal{L}_{\text{SFT}} = -\frac{1}{|\mathcal{D}_{\text{SFT}}|} \sum_{C \in \mathcal{D}_{\text{SFT}}} \log \pi_\theta\big(\text{COT}_C \mid C\big), \tag{11}$$

where π_θ denotes the LLM policy parameterized by θ.

3.2.2 Policy Optimization

After supervised fine-tuning, we further refine the model using GRPO. During this stage, the model takes as input an anchor node and a candidate set of nodes, and generates multiple candidate multi-hop paths. These generated paths are evaluated using a reward function, which guides the model to produce socially plausible and cross-community connections.

Path Length Preference To discourage degenerate single-hop predictions, we reward paths whose length approaches the maximum hop limit

$$R_{\text{len}} = \frac{|C| - 1}{6}. \tag{12}$$

Social Homophily To preserve semantic coherence, we encourage nodes along the path to remain close to the anchor user in the profile embedding space

$$R_{\text{homo}} = \frac{1}{|C|} \sum_{i=0}^{|C|-1} \cos(\mathbf{x}_{v_0}, \mathbf{x}_{v_i}). \tag{13}$$

Influence-aware Traversal To promote shortcut formation across communities, we reward paths that traverse structurally influential nodes

$$R_{\text{inf}} = \frac{1}{|C| - 2} \sum_{i=1}^{|C|-2} \tilde{d}^-(v_i). \tag{14}$$

The final reward for GRPO is computed as

$$R_{\text{GSI}} = R_{\text{len}} + R_{\text{homo}} + R_{\text{inf}}, \tag{15}$$

which guides the model to generate socially coherent multi-hop connections during training.

4 GraphMind-Botnet

In this section, we introduce GraphMind-Botnet, a simulated social network constructed by autonomous GraphMind agents. The resulting network exhibits structural properties that closely resemble those of real-world human social graphs, including community organization and small-world connectivity.

4.1 Basic Framework

User Roles. GraphMind agents are assigned user roles to simulate realistic online profiles and individual preferences. Usernames and profile descriptions are generated by an LLM and stored in a centralized database. Demographic attributes, including age, gender, education level, and geographic location, are sampled to match the empirical distributions observed in the TwiBot-22 training dataset. This design ensures that the synthetic population reflects realistic user diversity at the attribute level.

MCP-based Interaction Execution. All external social actions, such as following, liking, and commenting, are abstracted as callable operations under the Model Context Protocol (MCP) (Hou et al., 2025). By encapsulating environment interactions into standardized MCP interfaces, agents are able to interact with the simulated social platform in a structured and modular manner. This abstraction also facilitates extensibility, allowing additional interaction types or platform rules to be incorporated without modifying the agent logic.

4.2 Botnet Construction

GraphMind agents autonomously construct the simulated social network through heuristic reasoning and iterative interaction with the environment. The construction process consists of two tightly coupled components: the generation of follow relationships to establish network topology, and the refinement of interaction behaviors to populate the network with realistic social activity.

4.2.1 Follow Relationship Generation

The formation of follow relationships proceeds by first establishing cohesive intra-community structures and subsequently introducing cross-community connections to ensure global connectivity. To this end, bots are partitioned into N communities based on the profile embeddings of LLM-driven agents, such that agents within the same community share similar demographic characteristics and content preferences. Guided by the language-style similarity principle (Kovacs and Kleinbaum, 2020), bots within each community naturally form dense and cohesive subgraphs, capturing the homophilic tendencies commonly observed in online social networks.

Beyond intra-community connectivity, real-world social platforms such as Facebook and X are known to form globally connected graphs that exhibit the six degrees of separation phenomenon (Ugander et al., 2011; Myers et al., 2014). To reproduce this property, GraphMind introduces multi-hop follow relationships between otherwise disconnected agents. Specifically, pairs of nodes without an existing path between them are randomly sampled, and the corresponding bots are prompted to infer plausible multi-hop follow chains based on the current network topology. These inferred paths are then materialized by invoking the network simulation module through MCP, which instantiates the missing follow edges. Repeating this process incrementally yields a structurally complex and well-connected botnet. The overall procedure is briefly summarized in Algorithm 1.

Table 1: The composition of the our dataset

Human	Bot	Edges	Edge Types	Communities
1,000	1000	41375	5	50

4.2.2 Interaction Behavior Refinement

Once the follow network has been established and exhibits small-world characteristics, agents engage in fine-grained social interactions with their one-hop neighbors to populate the network with heterogeneous behaviors. At this stage, each LLM-driven bot analyzes the profile attributes and inferred preferences of its neighbors, estimates the corresponding social tie strength. Then they generate diversified interaction behaviors that are consistent with this relationship. These interactions, including likes, comments, and other observable actions, are executed and recorded through the MCP interface.

Through this two-stage construction process, GraphMind-Botnet evolves into a socially coherent network in which both the structural topology and the interaction patterns closely mirror those of human-operated social platforms.

4.3 GraphMind Dataset

Composition. We aggregate the follower network and interaction records generated during simulation and integrate them with human nodes sampled from the TwiBot-20 dataset (Feng et al., 2021b) to construct the GraphMind Dataset. Table 1 summarizes its composition, including the number of human users, bots, edge types, and communities. A visualization of the following network is provided in Appendix. The bot network exhibits both dense intra-community structures and sparse yet meaningful inter-community links. These human-like characteristics enhance the realism of the dataset and increase its potential to confuse detection.

Model and Training. We adopt Qwen3 1.7B and Qwen3 8B (Yang et al., 2025) as the base LLMs for GraphMind agents. Both models provide a favorable trade-off between model capacity and computational efficiency, enabling scalable simulation of large social networks while retaining sufficient expressive power to capture complex social behaviors. Training and inference are performed on a single NVIDIA RTX 4090 GPU, with each network generation requiring approx-

Table 2: Bot detection performance across datasets (lower values indicate stronger evasion).

Dataset	Metric	Metadata-based				Text-based	Homo-GNN		Heter-GNN		
		AB	RF	DT	SVM	Wei *et al.*	GCN	GAT	BotRGCN	RGT	S-HGN
Cresci-15	Acc	96.9	96.0	94.7	95.4	94.78	98.2	97.7	96.3	97.2	96.9
	F1	95.5	95.6	94.6	95.3	84.25	96.0	97.0	96.3	96.5	95.9
Cresci-17	Acc	93.2	88.1	86.2	85.1	86.30	/	/	/	/	/
	F1	83.4	78.9	79.4	76.8	79.40	/	/	/	/	/
Twibot-20	Acc	85.9	85.4	81.1	85.7	78.26	76.8	83.2	86.8	87.4	85.9
	F1	84.6	83.9	80.2	84.8	76.5	75.3	81.9	86.6	85.7	84.7
MGTAB-22	Acc	92.5	90.1	88.1	87.7	/	85.2	87.4	89.6	92.1	90.4
	F1	88.6	87.8	85.7	86.3	/	78.8	84.3	87.2	90.4	87.7
Twibot-22	Acc	67.3	73.6	74.6	78.4	69.7	81.6	76.3	81.6	73.9	76.7
	F1	35.8	32.4	50.6	52.6	54.6	56.8	54.6	58.4	45.1	45.7
EvoBot	Acc	74.4	69.3	70.1	75.7	58.8	78.1	75.7	88.5	80.3	82.6
	F1	70.3	65.7	66.6	70.8	54.6	68.4	61.5	83.2	77.8	80.3
ChatGPT-3.5											
OASIS	Acc	67.9	73.4	66.8	70.4	68.3	80.5	70.7	82.7	76.3	85.6
	F1	51.6	44.5	50.6	61.3	59.2	79.8	62.5	81.3	75.6	80.5
BotSim-24	Acc	77.5	75.7	71.4	74.4	50.8	72.7	80.3	89.9	82.3	87.7
	F1	74.8	72.4	68.5	69.8	50.4	50.5	73.1	86.7	76.4	83.1
Qwen3 1.7B											
GraphMind Dataset	Acc	74.4	**69.3**	70.1	72.7	50.8	61.9	76.8	72.4	73.5	70.7
	F1	70.3	65.7	66.6	70.8	44.6	46.5	67.3	67.6	65.8	60.5
Qwen3 7B											
GraphMind Dataset	Acc	**71.3**	69.6	**65.7**	**68.4**	**46.7**	**60.1**	**75.7**	**70.5**	**69.6**	**69.1**
	F1	68.8	66.6	63.0	64.2	46.4	58.4	61.5	62.6	65.3	68.4

Note: Lower values correspond to better evasion performance and reduced detectability.

imately 18 hours. Both modules of the GraphMind framework are trained using datasets of 3k samples. The current GraphMind Dataset is generated based on training and inference using the Qwen-3 1.7B model. Detailed model architectures and training hyperparameters are provided in Appendix D.

5 Experiment

5.1 Experimental Settings

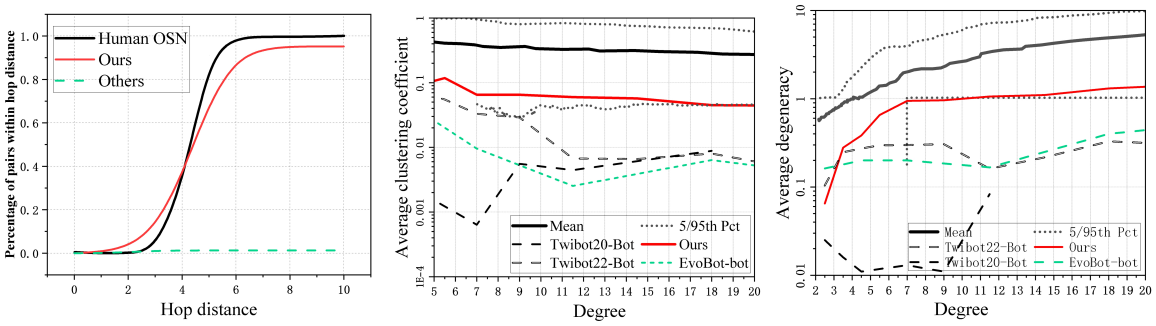
Datasets. We evaluate the effectiveness of the GraphMind Dataset by comparing it against existing social bot datasets and simulation frameworks. Traditional benchmarks, including Cresci-15 (Cresci et al., 2015), Cresci-17 (Cresci et al., 2017), TwiBot-20 (Feng et al., 2021b), TwiBot-22 (Feng et al., 2022a), and MGTAB-22 (Shi et al., 2025), serve as references for performance comparison. To fairly reproduce other LLM-driven social bot simulations, we follow the original authors' open-source implementations: OASIS (Yang et al., 2024) is simulated with 1,000 agents over 50 interaction rounds, whereas BotSim (Qiao et al., 2025) and EvoBot (Kong et al., 2025) directly use the datasets released by the original works. Since OASIS only contains bot samples, we supplement it with 1,000 human nodes to ensure a comparable human–bot mixture with GraphMind Dataset.

Detectors. We consider four categories of detection algorithms: (1) Meta-based detectors, including AdaBoost (Zhu et al., 2009), Random Forest (Yang et al., 2020), Decision Tree (DT) (Lepping, 2018), and SVM (Boser et al., 1992); (2) Text-based detectors (Wei and Nguyen, 2019); (3) Homogeneous GNN detectors, such as GCN (Kipf, 2016) and GAT (Veličković et al., 2017); and (4) Heterogeneous GNN detectors, including S-HGN (Lv et al., 2021), BotRGCN (Feng et al., 2021c), and RGT (Feng et al., 2022b). This setup allows a thorough evaluation of both traditional and LLM-driven botnets under diverse detection settings. Detection models are trained using an 80/20 split on the respective training set. Detectors are trained on TwiBot-22 and evaluated on each LLM-driven social bot simulation (OASIS, BotSim, EvoBot, and GraphMind). This cross-dataset setting is deliberately adopted to quantify the detectability gap between simulated bot networks and real human networks.

5.2 Experiment Results

5.2.1 Bot Detection Evasion Performance

Table 2 reports the evasion performance of GraphMind social bot against a wide range of bot detection methods. Since lower detection accuracy or F1 score corresponds to stronger evasion, the results show that GraphMind dataset consistently degrades the effectiveness of existing detectors. Notably, both homogeneous and heterogeneous GNN-based models suffer substantial performance drops (F1 scores of 60-70% compared to 85-95% on traditional datasets), indicating that GraphMind social bots present significant challenges for structure-aware detectors. While detection remains feasible, the explicit modeling of network topology substantially increases the difficulty of bot identification. This observation suggests that explicitly enhancing network-level realism in LLM-driven social simulations significantly increases the challenge of bot detection. Notably, even with a lightweight base model (Qwen3-1.7B), GraphMind dataset reduces GNN detection accuracy to 61.9% (F1: 46.5%), representing a substantial improvement in evasion capability compared to prior LLM-based botnet



(a) Hop Distances (b) Clustering Coefficient (c) Average Degeneracy

Figure 3: Structural property analysis of different network

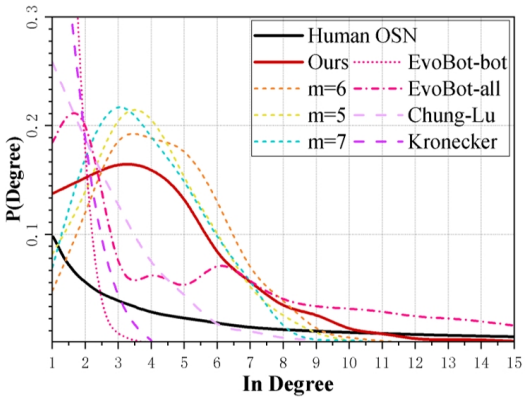


Figure 4: Degree distributions

simulations (OASIS: 80.5% acc, BotSim: 72.7% acc).

5.2.2 Network Structure Analysis

To further analyze the differences between our botnet, human networks, and prior methods, we qualitatively evaluate their structural commonalities using standard network analysis metrics.

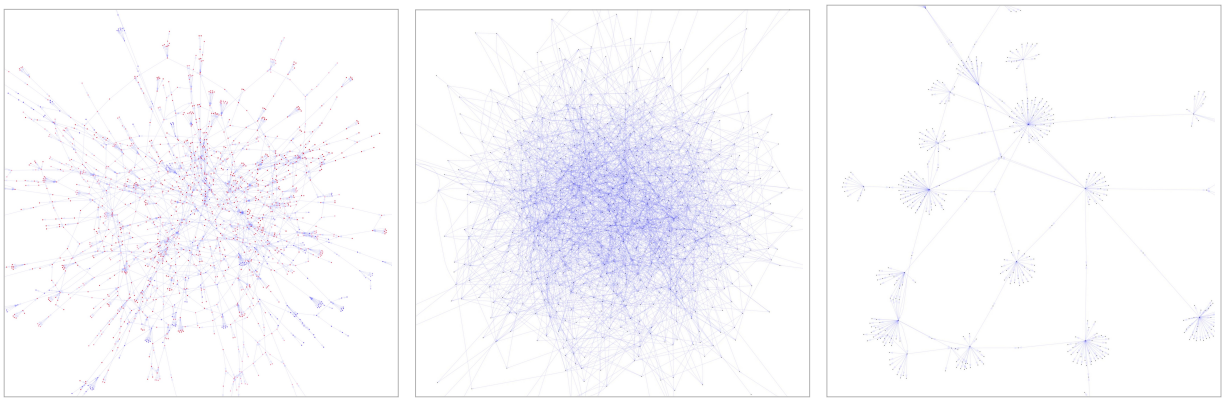
Path Length. The distribution of shortest-path distances between node pairs is a fundamental macroscopic property of social networks. Following prior work on OSNs (Ugander et al., 2011), we analyze follow-path lengths and compare our method with existing social bot simulation approaches. We characterize node-to-node proximity using the neighborhood function $P(h)$, defined as the fraction of node pairs (u, v) such that u is reachable from v by a path of length at most h. As shown in Figure 3a, human social networks exhibit an average pairwise distance of 4.7, with approximately 92% of user pairs connected within five hops (Ugander et al., 2011). Our generated botnet closely matches this empirical pattern, achieving an average hop distance of 4.77, with 90% of node pairs connected within six hops.

In contrast, existing social bot simulation methods, including OASIS (Yang et al., 2024), EvoBot (Kong et al., 2025), and BotSim (Qiao et al., 2025), produce extremely sparse graphs with poor global connectivity. Most nodes in these baselines remain isolated, resulting in overall reachability below 2% and preventing meaningful path-length statistics.

Clustering Coefficient and Degeneracy. We evaluate the structural realism of generated social graphs using the local clustering coefficient and graph degeneracy, which jointly capture local cohesion and densely connected cores. Following prior work (Leskovec et al., 2008), directed follow edges are treated as undirected via a standard Facebook-style neighbor projection. Figure 3b plots the average clustering coefficient as a function of node degree, with the empirical mean and 5th/95th percentiles of Facebook networks as reference. For degrees below 20, human social networks consistently exhibit clustering coefficients above 30%, and only GraphMind reliably falls within this empirical range, while other botnets deviate substantially.

We further analyze graph degeneracy, defined as the maximum non-empty k-core. As shown in Figure 3c, GraphMind consistently lies within the percentile range. Botnets from TwiBot20/22 and recent LLM-driven simulations such as OASIS (Yang et al., 2024) and BotSim (Qiao et al., 2025) yield near-zero values on both metrics, indicating fragmented graphs with limited core structure. All metrics are computed primarily for nodes with degree < 20, which is appropriate for small synthetic graphs and enables reliable characterization of local structural patterns.

Network visualizations. To further validate the accuracy of our data analysis, we visualize several representative botnet and human follow networks. Figures 5a present the human follow network in TwiBot20, which exhibits a mesh-like structure characterized by a densely connected core and progressively sparser peripheral regions. In contrast, existing social bot simulations predominantly form a star-like topology, with a single dominant core and a large number of isolated nodes, as shown in Figure 5c. By comparison, the network constructed by our method more closely resembles the mesh-like topology observed in human networks (Figure 5b), introducing a substantial number of plausible follow edges and resulting in a large, well-connected component rather than fragmented network structures.



(a) Visualization of human follow network in TwiBot20

(b) Visualization of bot follow network in GraphMind (ours)

(c) Visualization of bot follow network in Evobot (part)

Figure 5: Network visualizations comparing human and bot follow networks across different datasets.

Table 3: Ablation Study of GraphMind Modules on Detection Evasion (%).

Methods	S-HGN		BotRGCN		RGT	
	Acc	F1	Acc	F1	Acc	F1
All						
Qwen3-1.7B	70.7	60.5	72.4	67.6	73.5	65.8
Qwen3-8B	69.1	68.4	70.5	62.6	69.6	65.3
w/o FIM						
Qwen3-1.7B	78.9	69.5	79.8	74.2	80.5	73.1
Qwen3-8B	77.2	75.1	78.4	70.5	77.9	72.8
w/o FIM+GSI						
Qwen3-1.7B	86.4	80.2	87.5	83.4	88.1	81.0
Qwen3-8B	85.8	84.5	86.2	81.1	85.5	82.3

*Note: **FIM** denotes Fine-Grained Interaction Modeling, **GSI** refers to Graph-Augmented Social Inference. Lower Accuracy and F1 indicate stronger bot evasion.*

5.3 Experimental Analysis

5.3.1 Ablation Experiments

The fine-grained interaction modeling module enables a richer action repertoire in bot–bot interactions. As shown by the ablation study in Table 3, removing this component leads to a substantial increase in detection accuracy for heterogeneous GNN-based methods. This is because heterogeneous GNNs are designed to capture diversity in node-level operations, and the absence of fine-grained interaction modeling causes the botnet to degenerate structurally. Specifically, agents exhibit action homogeneity, repeatedly executing a limited set of operations (e.g., commenting), which makes the network easier to detect. In contrast, agents trained with this module dynamically select context-appropriate actions conditioned on tweet intent, resulting in behavior sequences that exhibit stronger contextual coherence and emotional consistency.

5.3.2 Comparison against simple graph generators

To compare multi-hop connections generated by LLM-driven agents with randomly constructed ones, we conduct a controlled experiment in which communities are connected either by agent inference or by random m-hop chains, where m denotes the path length. In the random setting, disconnected node pairs are repeatedly sampled and linked via $m-2$ randomly selected intermediate nodes until 97% of node pairs become reachable. We evaluate the resulting graphs using the in-degree distribution (Figure 4). Compared to random strategies, LLMs tend to prioritize and identify influential core nodes within communities, characterized by high in-degree centrality. As a result, the GraphMind-Botnet exhibits a heavy-tailed, power-law-like in-degree distribution with several high-impact nodes (Myers et al., 2014). Figure 4 also includes comparisons with classical graph generators, including the Chung–Lu model (Fasino et al., 2021) and the Kronecker graph model (Leskovec et al., 2010). GraphMind simultaneously preserves realistic long-tail behavior and uniformity consistent with real social networks. Notably, although the Evobo dataset appears long-tailed at the aggregate level, isolating bot nodes reveals that most bots have in-degree one and none exceed degree four, indicating a superficial rather than structural long-tail effect. Additional visualizations and statistics are provided in Appendix E.

6 Conclusion

GraphMind pioneers seamless integration of social graphs with LLM-powered bots via dual-stage training, enabling autonomous human-like topology construction. The GNN-Augmented Social Inference module incorporates small-world effects and homophily principles for multi-hop connection discovery, while the Fine-Grained Interaction Modeling module decouples behavioral operations from semantic generation to implement affectively coherent atomic actions via MCP protocol. The resultant botnet exhibits topological properties that closely resemble those of human networks, underscoring the need for resilient countermeasures against evolving LLM social bots.

7 Limitations

Current approaches exhibit several limitations. First, existing network generation methods struggle to scale to large-scale social networks. Due to the inherent context length constraints of large language models, when the number of agents exceeds approximately 2k, LLM-based social bots can no longer access complete global network information, leading to distorted or inconsistent link generation.

Second, current methods primarily operate at a macroscopic level by training LLMs to heuristically complete missing social links, rather than maintaining social relationships through fine-grained, agent-level interactions. In future work, we plan to enable LLM-driven social bots to actively establish and maintain accurate social connections at the micro level through continuous local interactions.

Finally, the evaluation of how closely agent-generated networks resemble real human social networks remains largely qualitative. More rigorous quantitative analyses are required to assess structural fidelity, which is crucial for studying information diffusion and large-scale collective behavior in simulated social systems.

References

Seyed Ali Alhosseini, Raad Bin Tareaf, Pejman Najafi, and Christoph Meinel. 2019. Detect me if you can: Spam bot detection using inductive representation learning. In *Companion proceedings of the 2019 world wide web conference*, pages 148–153.

Valerio Arnaboldi, Marco Conti, Massimiliano La Gala, Andrea Passarella, and Fabio Pezzoni. 2016. Ego network structure in online social networks and its impact on information diffusion. *Computer Communications*, 76:26–41.

Yuanchen Bei, Weizhi Zhang, Siwen Wang, Weizhi Chen, Sheng Zhou, Hao Chen, Yong Li, Jiajun Bu, Shirui Pan, Yizhou Yu, and 1 others. 2025. Graphs meet ai agents: Taxonomy, progress, and future opportunities. *arXiv preprint arXiv:2506.18019*.

Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152.

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

Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. 2017. The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. In *Proceedings of the 26th international conference on world wide web companion*, pages 963–972.

Shilpa Devalal and A Karthikeyan. 2018. Lora technology-an overview. In *2018 second international conference on electronics, communication and aerospace technology (ICECA)*, pages 284–290. IEEE.

Sabit Ekin. 2023. Prompt engineering for chatgpt: a quick guide to techniques, tips, and best practices. *Authorea Preprints*.

Dario Fasino, Arianna Tonetto, and Francesco Tudisco. 2021. Generating large scale-free networks with the chung–lu random graph model. *Networks*, 78(2):174–187.

Shangbin Feng, Zhaoxuan Tan, Herun Wan, Ningnan Wang, Zilong Chen, Binchi Zhang, Qinghua Zheng, Wenqian Zhang, Zhenyu Lei, Shujie Yang, and 1 others. 2022a. Twibot-22: Towards graph-based twitter bot detection. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Shangbin Feng, Zhaoxuan Tan, Herun Wan, Ningnan Wang, Zilong Chen, Binchi Zhang, Qinghua Zheng, Wenqian Zhang, Zhenyu Lei, Shujie Yang, and 1 others. 2022b. Twibot-22: Towards graph-based twitter bot detection. *Advances in Neural Information Processing Systems*, 35:35254–35269.

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

Shangbin Feng, Herun Wan, Ningnan Wang, Jundong Li, and Minnan Luo. 2021b. Twibot-20: A comprehensive twitter bot detection benchmark. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 4485–4494.

Shangbin Feng, Herun Wan, Ningnan Wang, and Minnan Luo. 2021c. Botrgcn: Twitter bot detection with relational graph convolutional networks. In *Proceedings of the 2021 IEEE/ACM international conference on advances in social networks analysis and mining*, pages 236–239.

Chen Gao, Xiaochong Lan, Zhihong Lu, Jinzhu Mao, Jinghua Piao, Huandong Wang, Depeng Jin, and Yong Li. 2023. S3: Social-network simulation system with large language model-empowered agents. *arXiv preprint arXiv:2307.14984*.

Buyun He, Yingguang Yang, Qi Wu, Hao Liu, Renyu Yang, Hao Peng, Xiang Wang, Yong Liao, and Pengyuan Zhou. 2024. Botdgt: Dynamicity-aware social bot detection with dynamic graph transformers. *arXiv preprint arXiv:2404.15070*.

Xinyi Hou, Yanjie Zhao, Shenao Wang, and Haoyu Wang. 2025. Model context protocol (mcp): Landscape, security threats, and future research directions. *ArXiv*.

TN Kipf. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Fanqi Kong, Xiaoyuan Zhang, Xinyu Chen, Yaodong Yang, Song-Chun Zhu, and Xue Feng. 2025. Enhancing llm-based social bot via an adversarial learning framework. *arXiv preprint arXiv:2508.17711*.

Balazs Kovacs and Adam M Kleinbaum. 2020. Language-style similarity and social networks:. *Psychol Sci*, (2).

Joachim Lepping. 2018. Wiley interdisciplinary reviews: Data mining and knowledge discovery.

Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. 2010. Kronecker graphs: an approach to modeling networks. *Journal of Machine Learning Research*, 11(2).

Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney. 2008. Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th international conference on World Wide Web*, pages 695–704.

Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang. 2021. Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 1150–1160.

Stanley Milgram and 1 others. 1967. The small world problem. *Psychology today*, 2(1):60–67.

Seth A Myers, Aneesh Sharma, Pankaj Gupta, and Jimmy Lin. 2014. Information network or social network? the structure of the twitter follow graph. In *Proceedings of the 23rd international conference on world wide web*, pages 493–498.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.

Joon Sung Park, Lindsay Popowski, Carrie Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2022. Social simulacra: Creating populated prototypes for social computing systems. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, pages 1–18.

Boyu Qiao, Kun Li, Wei Zhou, Shilong Li, Qianqian Lu, and Songlin Hu. 2025. Botsim: Llm-powered malicious social botnet simulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 14377–14385.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and 1 others. Lora: Low-rank adaptation of large language models.

Shuhao Shi, Kai Qiao, Zihao Liu, Jie Yang, Chen Chen, Jian Chen, and Bin Yan. 2025. Mgtab: A multi-relational graph-based twitter account detection benchmark. *Neurocomputing*, page 130490.

Shazia Tabassum, Fabiola SF Pereira, Sofia Fernandes, and João Gama. 2018. Social network analysis: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(5):e1256.

Yanchao Tan, Zihao Zhou, Hang Lv, Weiming Liu, and Carl Yang. 2023. Walklm: A uniform language model fine-tuning framework for attributed graph embedding. *Advances in neural information processing systems*, 36:13308–13325.

Johan Ugander, Brian Karrer, Lars Backstrom, and Cameron Marlow. 2011. The anatomy of the facebook social graph. *arXiv preprint arXiv:1111.4503*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Feng Wei and Uyen Trang Nguyen. 2019. Twitter bot detection using bidirectional long short-term memory neural networks and word embeddings. In *2019 First IEEE International conference on trust, privacy and security in intelligent systems and applications (TPS-ISA)*, pages 101–109. IEEE.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Qi Wu, Yingguang Yang, Buyun He, Hao Liu, Renyu Yang, and Yong Liao. 2024. Botscl: Heterophily-aware social bot detection with supervised contrastive learning. In *International Conference on Pattern Recognition*, pages 53–68. Springer.

Xiao-Kun Wu, Min Chen, Wanyi Li, Rui Wang, Limeng Lu, Jia Liu, Kai Hwang, Yixue Hao, Yanru Pan, Qingguo Meng, and 1 others. 2025. Llm fine-tuning: Concepts, opportunities, and challenges. *Big Data and Cognitive Computing*, 9(4):87.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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

Ziyi Yang, Zaibin Zhang, Zirui Zheng, Yuxian Jiang, Ziyue Gan, Zhiyu Wang, Zijian Ling, Jinsong Chen, Martz Ma, Bowen Dong, and 1 others. 2024. Oasis: Open agent social interaction simulations with one million agents. *arXiv preprint arXiv:2411.11581*.

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and 1 others. 2023. Sotopia: Interactive evaluation for social intelligence in language agents. *arXiv preprint arXiv:2310.11667*.

Ji Zhu, Hui Zou, Saharon Rosset, Trevor Hastie, and 1 others. 2009. Multi-class adaboost. *Statistics and its Interface*, 2(3):349–360.

A Related work

A.1 Social Simulation Based on LLM.

Simulation of human behavior based on large language models (LLMs) has recently emerged as a prominent research frontier. The Smallville sandbox world enables natural language interaction with a town composed of twenty-five agents, demonstrating both believable individual behaviors and emergent social dynamics (Park et al., 2023). SimReddit introduces an LLM-driven simulation platform to study intentional social interactions (Park et al., 2022). S³ (Gao et al., 2023) combines Markov chains with LLMs to model opinion dynamics, while SOTOPIA proposes a framework for evaluating social intelligence (Zhou et al., 2023). Moving toward online social network (OSN) simulation, OASIS implements a general and scalable social media simulator to investigate large-scale collective phenomena and behaviors (Yang et al., 2024). BotSim designs an LLM-driven malicious social bot network simulator (Qiao et al., 2025), and EvoBot bypasses co-evolving detection systems through human-like expression strategies (Kong et al., 2025).

Despite these advances, existing frameworks primarily focus on refining LLMs' ability to imitate individual-level actions, while largely overlooking the explicit modeling of*social links between agents, such as follow or friendship relations. Under large-scale social simulation settings, such relational structures are typically assumed or manually specified, which is neither realistic nor scalable.

A.2 GNN-based Bot Detection

Graph-based detection methods have demonstrated strong effectiveness against bot evasion, particularly on benchmark datasets that incorporate relational graph information. Early work introduced Graph Convolutional Networks (GCNs) for bot detection by jointly modeling node attributes and network topology (Ali Alhosseini et al., 2019). Subsequent studies proposed self-supervised GCN-based frameworks to enhance representation learning (Feng et al., 2021a), as well as relational GCN architectures that support multi-relational aggregation in heterogeneous social graphs (Feng et al., 2021c).

However, these methods become increasingly inadequate as LLM-based social simulations evolve sophisticated evasion capabilities through authentic behavioral emulation and network construction, necessitating novel detection algorithms for advanced LLM-driven tactics.

B Ethics Statement

Following the original data usage terms, we collect and process data from the publicly available TwiBot-22 and TwiBot-20 datasets. To mitigate privacy risks, we employ prompt-based large language models to remove personally identifiable information from the original text, such as phone numbers and email addresses.

Similar to other LLM -based social simulation frameworks, GraphMind agents and the resulting GraphMind-Botnet can facilitate research on information diffusion and collective behavior by enabling the construction of more realistic social networks. However, the high structural fidelity of such agent-generated networks to real human social networks also introduces potential misuse risks, as they could be exploited by malicious actors to construct deceptive social botnets. To address this concern, we implement a strict application and review process for code and model parameter access, ensuring that all usage is limited to legitimate research purposes.

While GraphMind demonstrates the potential to evade existing social bot detection methods, particularly graph neural network (GNN)-based approaches, we emphasize that our work focuses on foundational modeling rather than deployment. The ethical implications of such capabilities must be carefully considered. As part of future work, we plan to investigate next-generation social bot detection mechanisms that are better suited to identifying LLM-based social bots and coordinated botnets. One promising direction is to leverage large language models to analyze and decompose suspicious network structures: although a botnet may internally control its relational topology, it often remains structurally isolated from the broader social graph. Properties such as the construction mechanisms and volume of follow edges may provide useful signals for botnet-level detection.

These efforts are essential for establishing ethical guidelines, safeguards, and regulatory frameworks that mitigate potential risks. Future research should prioritize the development of transparency and protection mechanisms to ensure responsible use while supporting the formulation of robust governance standards.

C Method Details

GraphMind has the potential to advance human–AI interaction by enabling more realistic social network simulation, with downstream benefits for applications such as collective behavior modeling and social bot detection. In addition, the development of detection algorithms capable of accurately identifying LLM-driven social bots will play a critical role in distinguishing human-generated from machine-generated content, thereby supporting the responsible deployment of such technologies. Below, we provide additional methodological details that are not explicitly described in the main body of the paper.

C.1 Fine-Grained Interaction Modeling

C.1.1 Social strength

In our framework, relationship strength does not directly prescribe specific interaction types, but instead constrains the frequency distribution of different interaction behaviors. This design is inspired by ego network theory, which characterizes strong and weak ties in terms of interaction intensity, frequency, and social cost, rather than specific action semantics. Accordingly, high-strength relationships are associated with more frequent high-engagement actions (e.g., reposting or replying), whereas lower-strength or asymmetric relationships tend to favor low-cost interactions such as liking. We view this mapping as an operationalization of established ego network principles, translating abstract tie strength into observable interaction patterns on online social platforms.

Table 4: Relationship Circle Definitions

Level	Definition
Level 1 (Support clique)	Strongest ties; frequent contact
Level 2 (Sympathy group)	Close ties; frequent but not weekly contact
Level 3 (Affinity group)	Casual ties; occasional contact
Level 4 (Active network)	All active ties; at least yearly contact

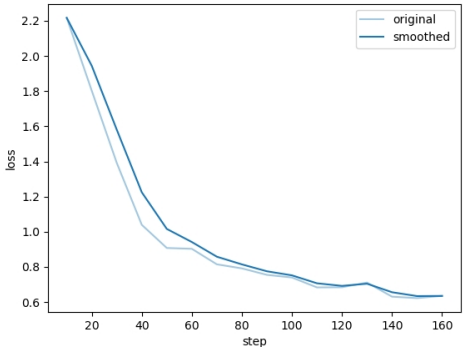


Figure 6: Loss function of FIM

Table 4 summarizes our definition of relationship strength. Specifically, we first annotate samples in the dataset by jointly considering interaction frequency between node pairs and profile similarity. During simulation, LLM-driven social bots assign target nodes to discrete relationship strength levels based on this definition, and subsequently generate interaction samples that follow the corresponding behavior frequency distributions.

C.1.2 Dataset and Prompt

We sample 3,000 node pairs with existing follow relationships from the TwiBot-22 dataset. Based on their historical interaction frequencies, we annotate the social relationship strength of one node with respect to the other. We then employ DeepSeek to complete the corresponding reasoning process for each annotated pair. Representative examples are shown in Table 7.

C.1.3 Training setting

Since large language models are not pretrained with explicit supervision for this task, we follow a widely adopted industry paradigm and apply supervised fine-

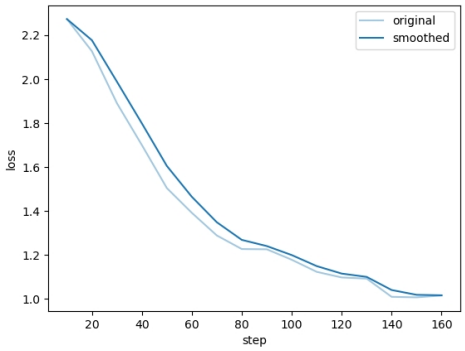


Figure 7: Loss function of GSI

tuning (SFT) to initialize the model. This cold-start stage provides stable behavior grounding before subsequent optimization. The training hyperparameters are reported in Table 5. The loss function are shown in Figure 6.

Hyperparameter	Value
Optimizer	AdamW
Learning rate	5×10^{-5}
Training epochs	5
LR scheduler	Cosine
Warmup steps	100
Random seed	42

Table 5: Key hyperparameters used for FIM.

C.2 Graph-Augmented Social Inference

C.2.1 Dataset and Prompt

We sample 3,000 seed nodes from the TwiBot-22 dataset and collect multi-hop social chains via random walks. For each pair of adjacent nodes, we leverage DeepSeek to infer the underlying rationale for the existence of follow relationships based on node attributes and structural features such as in-degree. The resulting step-by-step reasoning traces are then concatenated into long chains and used for training and optimization, with representative examples shown in Table 8.

C.2.2 Training setting

The training hyperparameters are reported in Table 6. The loss function are shown in Figure 7.

Hyperparameter	Value
Optimizer	AdamW
Learning rate	5×10^{-5}
Training epochs	5
LR scheduler	Cosine
Warmup steps	100
Random seed	42

Table 6: Key hyperparameters used for GSI.

C.3 GraphMind Dataset

For this dataset, we generate a botnet based on GraphMind agent, following the algorithm described below.

Algorithm 1 Multi-hop Follow Completion

Require: $G = \{V, E, X, Y\}$; connectivity threshold τ
Ensure: E

while NetworkConnectivity$(G) < \tau$ **do**
  Sample (u, v) with no path between u and v in G
  $C_u \leftarrow \text{Bot}(G,\ V_{\text{source}} = u,\ V_{\text{target}} = v)$
  $C_v \leftarrow \text{Bot}(G,\ V_{\text{source}} = v,\ V_{\text{target}} = u)$
  for each $C \in \{C_u, C_v\}$ with $C = [v_{h_0}, \ldots, v_{h_k}]$ **do**
    for $i = 0$ **to** $k - 1$ **do**
      $e \leftarrow (v_{h_i} \rightarrow v_{h_{i+1}})$
      if $e \notin E$ **then**
        $E \leftarrow E \cup \{e\}$
      end if
    end for
  end for
end while
return E

Field	Content
Instruction	You are an expert in social media behavior analysis. Your task is to infer the social relationship level between two users from an ego-network perspective, and then generate a list of interaction actions that follow realistic behavioral patterns. Follow the steps below: Step 1: Infer the most plausible social relationship level within the ego network. Step 2: Based on the inferred relationship strength, determine the expected interaction intensity and cost.
Input	<user_profile>: {PROFILE} <tweets>: {TWEETS}
Think	The target user belongs to my ego network with relationship level {RELATIONSHIP_LEVEL}. Based on this relationship strength and the observed profile and interaction context, I infer appropriate interaction intensity and engagement patterns.
Output	A list of interaction actions in the following structured format: <action> <type> like / retweet / comment / follow </type> <tweet_id> {TWEET_ID} </tweet_id>

Table 7: Prompt template used for FIM.

Field	Content
Instruction	You are an expert in social media behavior analysis. Given a source node and a target node, generate a multi-hop following path based on the input social network structure. First, select a set of potential intermediate nodes $\{n\}$ from the given network, and then predict and generate potential social links according to their profile information. Follow the steps below: Step 1: Select candidate intermediate nodes from the input graph based on profile similarity to the source node. Step 2: Sequentially infer plausible following relationships between adjacent nodes to form a multi-hop path. Requirements: 1. The path must contain ≥ 3 hops (Source $\rightarrow$ Mediator$_1$ $\rightarrow$ Mediator$_2$ $\rightarrow$ Target). 2. All mediator nodes must be selected from the given network information. 3. Path inference must be based on feature similarities (bio, age, education, location, etc.). 4. The connection rationale must be analyzed for each hop.
Input	<graph> <nodes> {users} </nodes> <edges> {follow_edges} </edges> </graph> <source_node> {user_A} </source_node> <target_node> {user_B} </target_node>
Think	<think> [Link 1] <user>1</user> follow <user>2</user> <reason>...</reason> [Link ..] </think>
Output	Based on the analysis of user profiles and social relationships, the most plausible multi-hop following path is: user_A $\rightarrow$ mediator_1 $\rightarrow$ mediator_2 $\rightarrow$ user_B.

Table 8: Prompt template used for GSI.

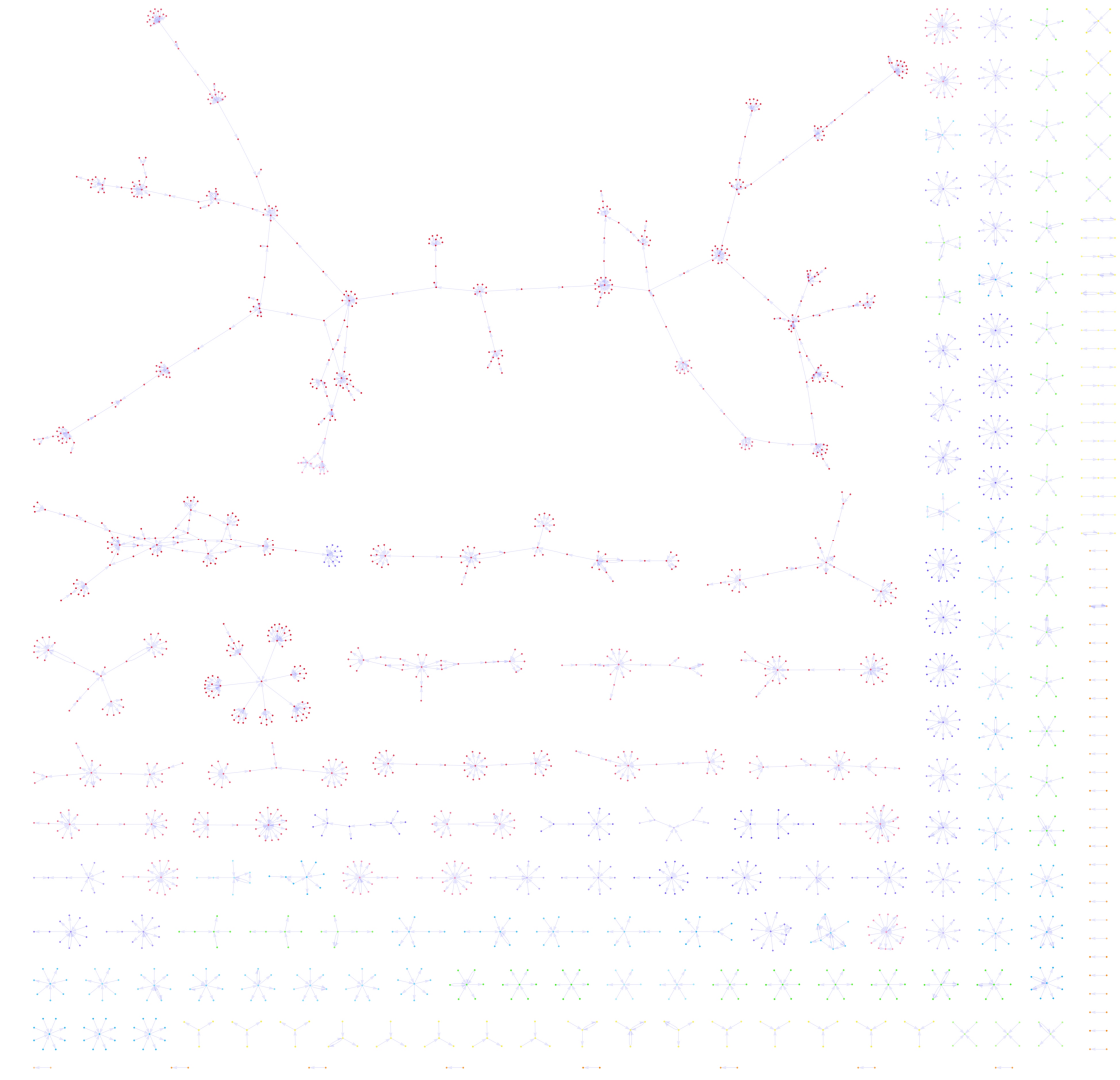


Figure 8: Botnet visualization of Twibot-20.

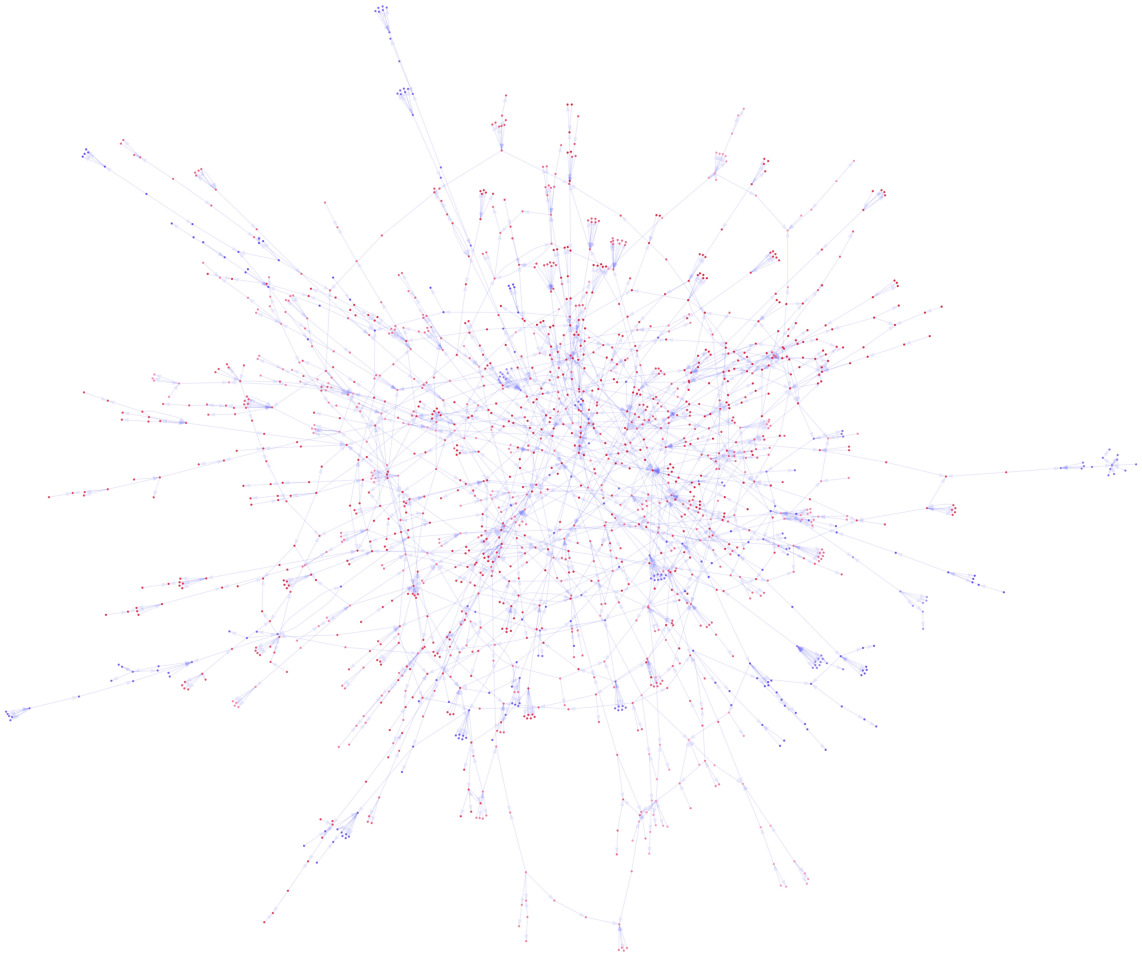


Figure 9: Human network visualization of Twibot-20.

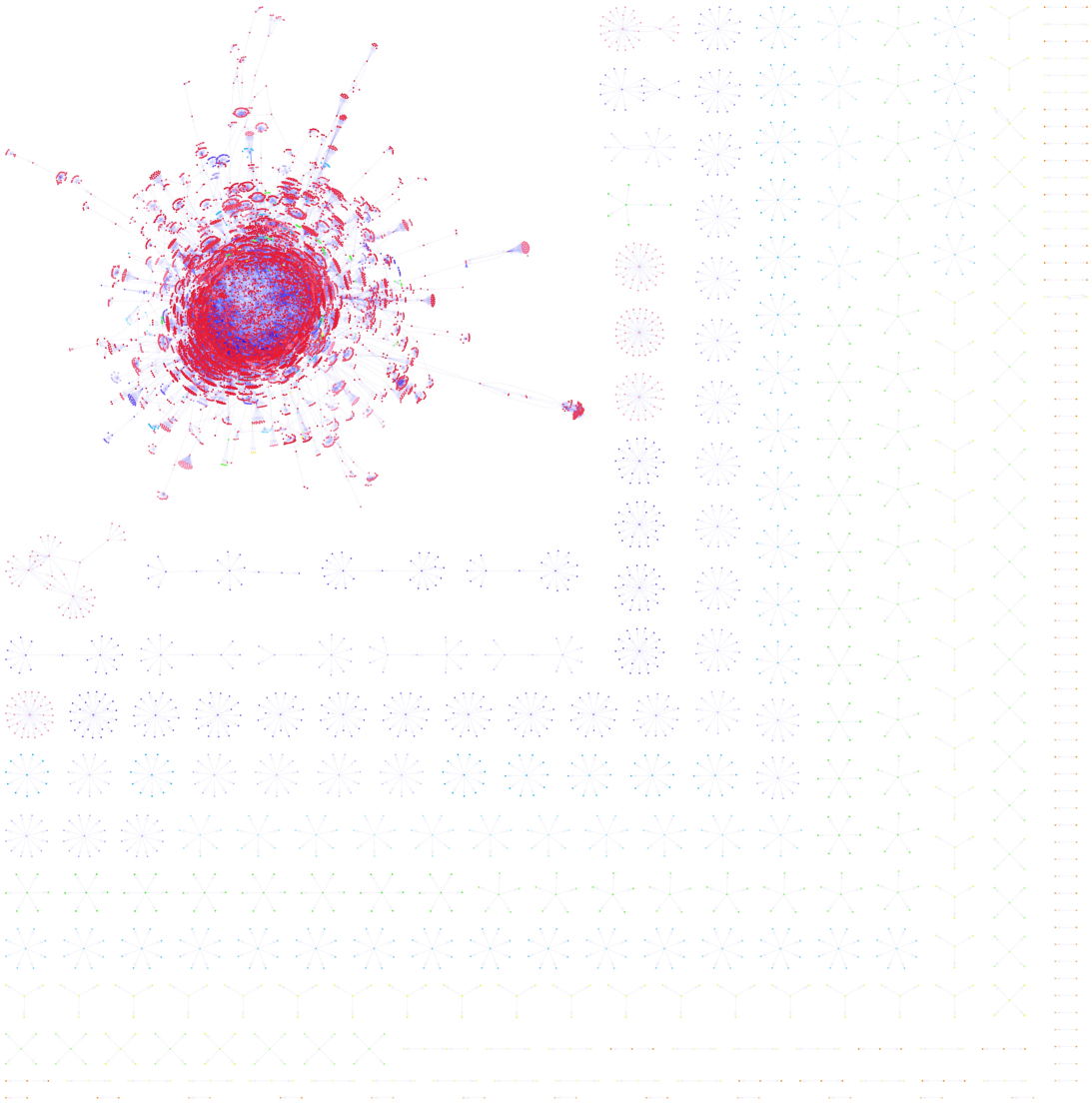


Figure 10: Botnet visualization of Twibot-22.

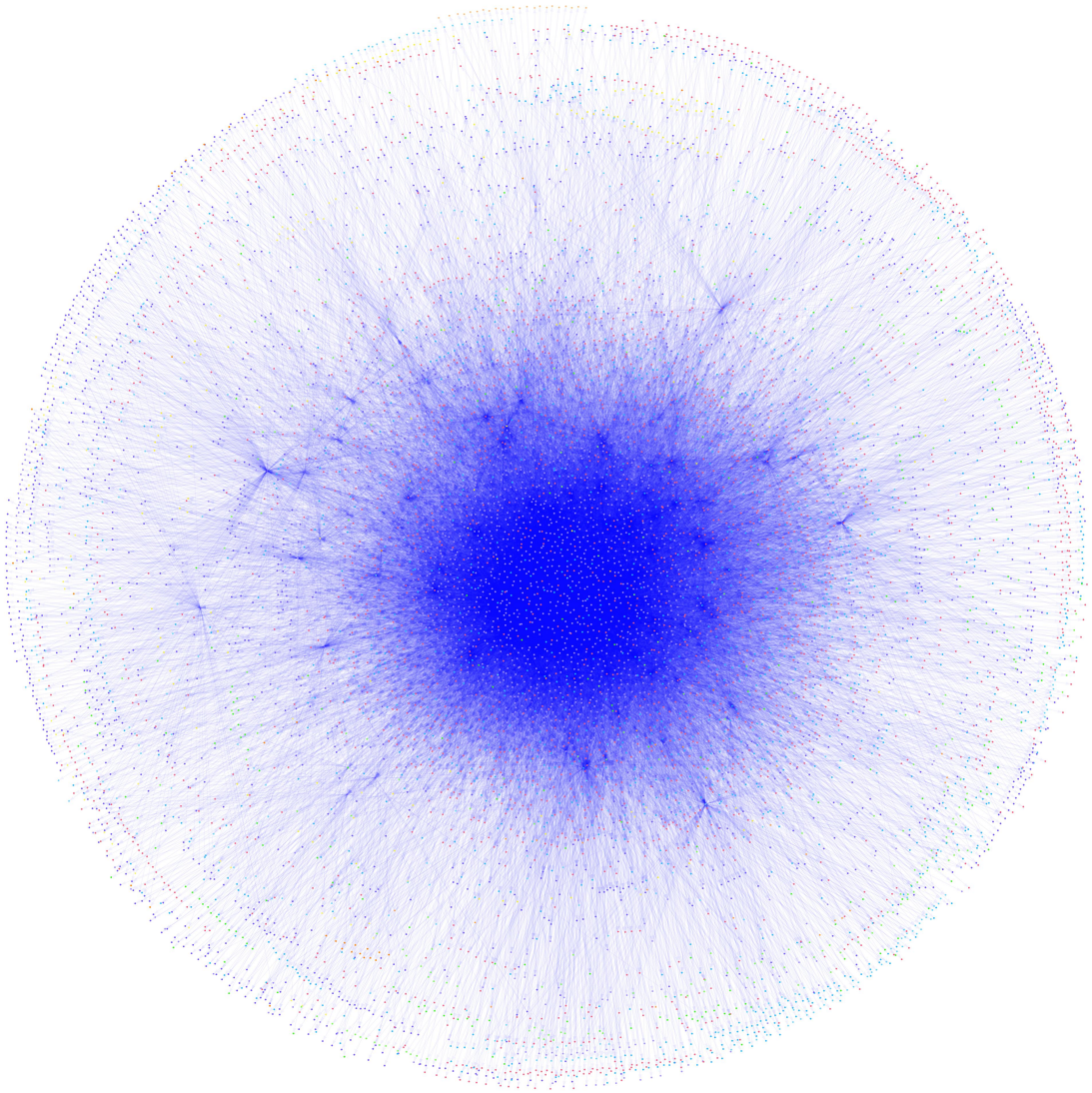


Figure 11: Visualization of EvoBot (human + bot).

Figure 12: Botnet Visualization of EvoBot.