Enhancing LLM-Based Social Bot via an Adversarial Learning Framework

Fanqi Kong[2,1], Xiaoyuan Zhang[2,1], Xinyu Chen[2], Yaodong Yang[2],
Song-Chun Zhu[1,2,3,4], Xue Feng[1*]

[1]State Key Laboratory of General Artificial Intelligence, BIGAI
[2]Peking University   [3]Tsinghua University
[4]PKU-WUHAN Institute for Artificial Intelligence
kfq20@stu.pku.edu.cn, fengxue@bigai.ai

Abstract

Developing Large Language Model (LLM) agents that exhibit human-like behavior, encompassing not only individual heterogeneity rooted in unique user profiles but also adaptive response to socially connected neighbors, is a significant research challenge. Social media platforms, with their diverse user data and explicit social structures, provide an ideal testbed for such investigations. This paper introduces EvoBot, an **Evo**lving LLM-based social **Bot** that significantly enhances human-like generative capabilities through a novel adversarial learning framework. EvoBot is initialized by Supervised Fine-Tuning (SFT) on representative data from social media and then iteratively refines its generation of sophisticated, human-like content via Direct Preference Optimization (DPO). This refinement is guided by feedback from a co-adapting **Detector** which concurrently improves its ability to distinguish EvoBot from humans, thereby creating an increasingly challenging learning environment for EvoBot. Experiments demonstrate that EvoBot generates content aligned with diverse user profiles, increasingly bypassing the co-adapting Detector through human-like expression. Moreover, it exhibits strong social responsiveness, more accurately modeling real-world opinion dynamics and information spread in multi-agent simulations. The framework also yields a more robust Detector, underscoring its broader utility for both advanced agent development and related detection tasks. The code is available at https://github.com/kfq20/EvoBot.

1 Introduction

A key aspiration in Large Language Models (LLMs) is to create autonomous agents that exhibit human-like behavior, moving beyond mere textual fluency to embody the richness of human interaction. Human-likeness in this context is profoundly multifaceted. It centrally involves individual heterogeneity, where an agent's expressions and actions are authentically rooted in unique personal characteristics, historical context, and specific profile attributes (Putnam, 2000; Tajfel, 1979). Simultaneously, it requires sophisticated social intelligence, reflecting how individuals dynamically perceive and respond to their socially connected neighbors and the broader social structure, leading to complex emergent phenomena such as opinion dynamics (Chuang et al., 2023; Ma et al., 2024), social influence (Abbas Naqvi et al., 2020; Peng et al., 2016) and information spread (like rumors, social-disease contagion) (Chopra et al., 2024; Bauch and Galvani, 2013; Feng et al., 2018, 2019). Endowing LLM-based agents with both this deep individual distinctiveness and adaptive social responsiveness remains a significant research challenge.

To instill multifaceted human-likeness in agents, learning environments must reflect real-world complexity. Social media platforms offer a uniquely rich setting for this: they combine detailed individual data (e.g., user profiles, posts, interaction histories) with complex social graphs (e.g., follower networks, communities). This confluence enables LLM agents to model both personalized expression and socially adaptive behavior (Gao et al., 2023, 2024; Yang et al., 2024; Kong et al., 2025). However, pre-trained LLMs often fail to capture the full range of human preferences, particularly those of marginalized communities, resulting in generic or biased outputs (Cheng et al., 2023; Chakraborty et al.; He et al., 2024b). Prompt engineering offers a lightweight method to guide model behavior without retraining, but its effectiveness comes with trade-offs: detailed prompts can improve authenticity, yet often reduce scalability and efficiency, echoing challenges seen in large-scale simulations. Fine-tuning provides stronger alignment with specific behaviors and personalities (Shao et al., 2023;

---

*Corresponding Author.

Ge et al., 2024), but it typically relies on human-labeled or high-quality synthetic data, which can introduce bias and limit generalizability (Chopra et al., 2024; Williams et al., 2023; Rafailov et al., 2024; Ouyang et al., 2022; Josifoski et al., 2023; Deng et al., 2023). This gap highlights the need for more effective learning paradigms, leading to our core research question:

How can LLMs learn from social media data to generate more human-like content?

To address this question, we propose a novel EvoBot (**Evo**lving LLM-based social **Bot**) through an adversarial learning framework, which is realized by leveraging the task of social bot detection, a well-established research area on social media (Feng et al., 2022, 2021a; Cresci et al., 2015, 2019). Specifically, a co-adapting **Detector** serves as an increasingly discerning 'adversary' providing dynamic feedback throughout the training of EvoBot, in contrast to traditional detectors trained on static datasets or not designed to address LLM-based bots (Yang et al., 2020; Feng et al., 2021b; Dialektakis et al., 2022; Wu et al., 2019).

EvoBot's learning begins with Supervised Fine-Tuning (SFT), where it is prompted with summarized user account information and details of their network neighbors to learn to generate tweet outputs that mimic genuine human communication patterns. Subsequently, EvoBot enters an iterative adversarial loop to refine its tweet generation for improved sophistication and authenticity via Direct Preference Optimization (DPO) (Rafailov et al., 2024). The crucial preference signals for DPO are derived from the Detector's classifications of EvoBot's generated tweets, guiding it towards outputs the Detector finds more human-like. The Detector, in turn, is regularly updated with EvoBot's evolving tweet outputs, ensuring a continually escalating challenge that propels EvoBot's learning.

The efficacy of our adversarial framework is substantiated by theoretical analysis and extensive empirical results, demonstrating EvoBot's advancements in achieving multifaceted human-likeness. At the individual level, EvoBot learns to generate content that reflects diverse user profiles and achieves a high degree of human-like expression, evidenced by its increasing success in bypassing the Detector. Furthermore, at the group level, EvoBot demonstrates its capacity for human-like social responsiveness, outperforming baseline models by more accurately replicating real-world group opinion and information spread. A key driver of EvoBot's improvements is the dynamic adversarial learning environment shaped by the co-adapting Detector, which continually retrains on EvoBot's evolving outputs. This process also improves the Detector's classification performance and generalization, making it a progressively stringent and adaptive benchmark for EvoBot's continued learning and development.

2 Methodology

The learning process of EvoBot includes two phases. **1)** SFT is conducted on real human data to pre-train EvoBot on the expressive habits, linguistic styles, and contextual preferences of community members. **2)** Adversarial learning is used, with both the EvoBot and the Detector iteratively trained. EvoBot's objective is to generate tweets that are most likely to be classified as human by the Detector, while the Detector aims to improve its accuracy in distinguishing between bots and humans. The following parts provide a problem formulation and a detailed description of all modules. Figure 1 provides an overview of this framework. And the learning process is detailed in Algorithm 1.

2.1 Problem Formulation

The social media dataset is modeled as a tuple $(\mathcal{V}, \{A_v\}, \{T_v\}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes, representing users, divided into two classes $\mathcal{H}$ (humans) and $\mathcal{B}$ (bots), i.e., $\mathcal{V} = \mathcal{H} \cup \mathcal{B}$. $\mathcal{E}$ is the set of directed edges, where $(u, v) \in \mathcal{E}$ indicates that user u follows user v. Each user $v \in \mathcal{V}$ is associated with two types of attributes: **Account Information** $A_v = \{a_1, a_2, \ldots, a_m\}$, which includes account features on Twitter, such as account creation time, number of followers, user description, and so on. **Tweets** $T_v = \{t_1, t_2, \ldots, t_n\}$, which represents a set of tweets posted by v.

The adversarial learning proceeds for K rounds. In the k-th round, **EvoBot**, represented by π_θ^k, generates tweets for a target user by integrating both the user's and their neighbors' information. Specifically, for user v, we use GPT-4o-mini (Hurst et al., 2024) (M_{sum}) to condense their account information A_v and historical tweets T_v into a concise summary $S_v = M_{\text{sum}}(A_v, T_v)$, which forms the first input. Similarly, the neighbor information is summarized as $S_{\mathcal{N}_v} = M_{\text{sum}}(A_{\mathcal{N}_v}, T_{\mathcal{N}_v})$. These, along with a task instruction $\mathcal{I}$, guide EvoBot to generate tweets that align with the target user's profile and
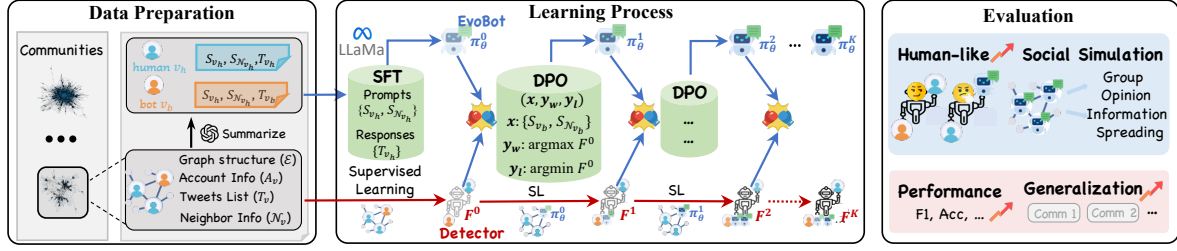
Figure 1: Overview of the EvoBot Framework, detailing its three core stages: **(1) Data Preparation:** Extraction of user and interaction data from social media communities. **(2) Learning Process:** EvoBot is initialized through SFT on human data. It then iteratively refines its human-like content generation via DPO, using feedback from a co-adapting Detector that serves as an evolving evaluative benchmark. **(3) Evaluation:** Assessment of EvoBot's enhanced multifaceted human-likeness, both at an individual level (e.g., through interaction with the Detector) and at a group level (via its ability to drive realistic group opinion and information spread simulations), alongside the Detector's resulting classification performance and generalization.

fit naturally into the community. The tweets are then generated as $T_v \propto \pi_\theta^k(T_v|\mathcal{I}, S_v, S_{\mathcal{N}_v})$.

Detector in the k-th round is defined as $F^k = \sum_{j=0}^k w^j f^j$, where $f : (\mathbf{A}, \mathbf{T}, \mathcal{E}) \to \mathbf{p}$ is the classifier trained in each round. Note that $F^0 = f^0$ is the base detector trained on the original dataset. $\mathbf{A} = \{A_1, A_2, \ldots, A_N\}, \mathbf{T} = \{T_1, T_2, \ldots, T_N\}$ represent account information and tweets for all N users, respectively. $\mathbf{p} = [p_1, p_2, \ldots, p_N]$ is the vector of probabilities, where $p_v \in [0, 1]$ is the probability that user v is classified as a human.

2.2 EvoBot

Supervised Fine-Tuning The SFT dataset is constructed by randomly selecting a subset of human users $\mathcal{H}_{\text{SFT}} \subseteq \mathcal{H}$. For each $v_h \in \mathcal{H}_{\text{SFT}}$, the prompt is $(\mathcal{I}, S_{v_h}, S_{\mathcal{N}_{v_h}})$. The reference response is the l tweets $T_{v_h} = \{t_{v_h,1}, t_{v_h,2}, \ldots, t_{v_h,l}\}$ sampled from the user's historical tweets T_{v_h}.

The objective of SFT is to minimize the discrepancy between the tweets generated by the base model of EvoBot π_θ^0 and the reference responses. This is achieved by optimizing the negative log-likelihood loss: $\mathcal{L}_{\text{SFT}} = -\frac{1}{|\mathcal{H}_{\text{SFT}}|} \sum_{v_h \in \mathcal{H}_{\text{SFT}}} \log \pi_\theta^0(T_{v_h}|\mathcal{I}, S_{v_h}, S_{\mathcal{N}_{v_h}})$.

Adversarial Learning with Detector EvoBot is trained to generate tweets that evade detection as bot-generated. A naive approach would be to use the Detector's output—the probability of being classified as human—as a scalar reward in reinforcement learning (Chen et al., 2024; Zhang et al., 2025; Kong et al., 2024). However, this faces issues like reward sparsity and unstable gradient estimation, leading to inefficient and suboptimal training (Cao et al., 2023; Zhang et al., 2024). So we use

DPO—a fine-tuning approach that directly uses the preference ordering in the data rather than training an additional reward model (Rafailov et al., 2024).

Specifically, in the k-th round of adversarial learning, N bot users $\{v_{b_i} \in \mathcal{B}, i = 1, \ldots, N\}$ are randomly sampled with replacement, and EvoBot generates C candidate responses $\{T_{v_{b_i}, c}\}_{c=1,\ldots,C}$ for each v_{b_i}. The Detector F^k then evaluates each candidate while keeping all other users' information fixed, calculating the probability that v_{b_i} is human for each response $T_{v_{b_i}, c}$, denoted as $F_{v_{b_i}, c}^k$.

To construct the DPO dataset $\mathcal{D}_{\text{DPO}} = \{x^i, y_w^i, y_l^i\}_{i=1}^N$, we let $x^i = (\mathcal{I}, S_{v_{b_i}}, S_{\mathcal{N}_{v_{b_i}}})$, $y_w^i = \arg\max_c F_{v_{b_i}, c}^k$, $y_l^i = \arg\min_c F_{v_{b_i}, c}^k$, where x^i is the input context, and y_w^i and y_l^i are the tweets with the highest and lowest probabilities of being classified as human, respectively. The loss function is
$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{x^i, y_w^i, y_l^i} \left[\log \sigma \left(\beta \log \frac{\pi_\theta^k(y_w^i|x^i)}{\pi_\theta^{k-1}(y_w^i|x^i)} - \beta \log \frac{\pi_\theta^k(y_l^i|x^i)}{\pi_\theta^{k-1}(y_l^i|x^i)} \right) \right]$,
where σ is the sigmoid function, and β is a hyperparameter controlling the deviation from the $k-1$'s version of EvoBot.

2.3 Detector

Features extraction Our detector employs a feature extraction approach inspired by (Feng et al., 2021b). More specifically, the classifier $f : (\mathbf{A}, \mathbf{T}, \mathcal{E}) \to \mathbf{p}$ takes as input account information $\mathbf{A}$, tweets $\mathbf{T}$, and the graph structure $\mathcal{E}$. The account information A_v includes numerical properties such as account creation time and number of followers, which are normalized for balanced scaling, as well as categorical properties like user description and verified status, represented using one-hot encoding for binary interpretability. The

textual data in T_v is embedded by RoBERTa (Liu, 2019) to capture semantic content. These features are processed through separate linear layers with LeakyReLU activations and then combined into a unified embedding. To incorporate the graph structure, we use an RGCN layer that aggregates relational information from the graph $\mathcal{E}$ based on the relation types. The resulting embeddings pass through fully connected layers with dropout regularization, producing a binary classification output that predicts whether a user is a bot or a human.

Supervised Learning In the k-th round of adversarial training, to obtain Detector F^k, all bot tweets in the dataset are replaced with outputs generated by EvoBot π_θ^{k-1} from the previous round. This modified dataset is then used to train the classifier f^k via supervised learning, using a cross-entropy loss to maximize classification accuracy.

2.4 Theoretical Analysis

In this subsection, we provide a theoretical analysis for our method from a more general view. We assume that data on social platforms can be represented as $(\mathbf{x}, \mathbf{y})$, where $\mathbf{x}$ denotes various user attributes, such as age, gender, occupation, interests, etc., sampled from the marginal distribution $q(\cdot)$ of the entire community. Meanwhile, $\mathbf{y} \sim \pi_{\mathcal{H}}(\cdot|\mathbf{x})$ represents the user's activities on the platform, such as posting tweets, retweeting, and liking, where $\pi_{\mathcal{H}}$ is the decision model of real humans in the community. Similarly, π_θ denotes EvoBot's model.

Detector is trained to maximize the probability of correctly classifying real and fake samples:

$$F = \arg\max \mathbb{E}_{\mathbf{x}\sim q(\cdot),\mathbf{y}\sim\pi_{\mathcal{H}}(\cdot|\mathbf{x})}[\log F(\mathbf{x},\mathbf{y})] \\ +\mathbb{E}_{\mathbf{x}'\sim q'(\cdot),\mathbf{y}'\sim\pi_\theta(\cdot|\mathbf{x}')}[\log(1 - F(\mathbf{x}',\mathbf{y}'))] \quad (1)$$

Here, the inputs $\mathbf{x}' \sim q'(\cdot)$ for EvoBot are distinguished from $q(\cdot)$, indicating that the input information received by EvoBot may come from a different distribution than the input received by real humans.

Considering the construction method of our DPO dataset in Section 2.2 and referring to Rafailov et al. (2024), the optimization objective of EvoBot is:

$$\pi_\theta = \arg\min \mathbb{E}_{\mathbf{x}'\sim q'(\cdot),\mathbf{y}'\sim\pi_\theta(\cdot|\mathbf{x}')}[1 - \log F(\mathbf{x}',\mathbf{y}')] \\ +\beta\mathbb{E}_{\mathbf{x}\sim q(\cdot),\mathbf{x}'\sim q'(\cdot)}[KL(\pi_{\mathcal{H}}(\cdot|\mathbf{x})\|\pi_\theta(\cdot|\mathbf{x}'))]. \quad (2)$$

Theorem 1. *If $q'(\mathbf{x}) = q(\mathbf{x})$, then under the iterative training process of the detector and generator with the optimization objective (1) and (2), the global optimum is achieved when $\pi_\theta = \pi_{\mathcal{H}}$.*

The proof is provided in Appendix A.

3 Experiment Setup

3.1 Dataset

We use TwiBot-22 (Feng et al., 2022), a graph-based Twitter dataset that includes one million users, nearly one hundred million tweets, and various relational data. In this dataset, we represent users as nodes and model follower-followee relationships as directed edges in a graph. Given the impracticality of training EvoBot on the entire dataset due to its size and complexity, we divide the network into smaller, more manageable communities using the *Louvain* community detection method (Blondel et al., 2008), identifying 12 highly connected and representative communities. These communities exhibit diverse network topologies (e.g., star-shaped, mesh-like structures), support multiple languages, and focus on a variety of topics. See Appendix B for data details and the preprocessing.

3.2 Models and Trainings

EvoBot is based on Llama-2-7b-chat (Touvron et al., 2023). For fine-tuning, we use the `transformers` and `trl` libraries to implement SFT and DPO. And we apply low-rank adaptation (LoRA) (Hu et al., 2021) using the `peft` library.

Training and inference are performed on 8 NVIDIA RTX 3090 GPUs, with each community requiring approximately 10 hours. EvoBot runs $K = 4$ iterations for adversarial learning, using 1024-sample datasets for both SFT and DPO. Detector is trained with an 8:1:1 split for training, validation, and test sets, with performance evaluated on the test set. Model architectures and training hyperparameters are detailed in the Appendix C.

3.3 Simulation Framework

We use the open-source social media simulation framework HiSim (Mou et al., 2024b) to analyze the response dynamics of EvoBot and baseline models as users react to trigger events, focusing on group opinion and information spread. Since EvoBot is designed for tweet generation, we simplify the simulation by excluding user actions like likes and retweets. EvoBot directly controls human accounts from the dataset to simulate user behavior. At each step, the input includes the prompt from Section 2.2, the current trigger event, recent past events, and the latest tweets from followed users.

4 Results

This section presents two sets of experiments. First, we assess EvoBot's enhanced generation of individually authentic content, alongside the co-adapting Detector that serves as an increasingly stringent benchmark. Second, we evaluate EvoBot's capacity for human-like social responsiveness at a group level through simulations of real-world group opinion and information spread.

4.1 Individual-Level Human-Likeness

To evaluate the impact of adversarial training on both EvoBot and the Detector, Figure 2 presents the classification performance in terms of F1-score and Accuracy (averaged across 12 different communities) across various training iterations. The use of both metrics is crucial due to the class imbalance between positive (human) and negative (bot) samples in the dataset. In the heatmaps, the cell at Detector F^i (row i) and EvoBot π_θ^j (column j) quantifies the result of F^i classifying bots that have been replaced with outputs from π_θ^j.

These average metrics highlight the co-adaptive learning dynamics: the 'Avg' raw for EvoBot shows a steady decline in average F1-score (from 0.770 for π_θ^0 to 0.452 for π_θ^4), indicating that later versions become increasingly evasive; meanwhile, the 'Avg' column for the Detector reveals progressively stronger performance across all EvoBot versions as training advances. However, performance is not strictly monotonic across all pairings—for instance, Detector F^2 may underperform F^1 against an earlier EvoBot such as π_θ^1, possibly due to overfitting on weaker or less diverse early outputs. Nevertheless, the overall trend reflects co-adaptive progress: EvoBot becomes more evasive and natural, while the Detector grows more robust—a pattern that also holds when using Llama-3-8B-Instruct (AI@Meta, 2024) as EvoBot's base, as shown in Appendix C.1, Figure 6. Furthermore, the adversarial training exhibits signs of convergence, as indicated by the diagonal F1-scores tracking Detector F^k against its counterpart EvoBot π_θ^k ($0.851 \rightarrow 0.792 \rightarrow 0.652 \rightarrow 0.558 \rightarrow 0.550$), which stabilize from $k = 3$ onward.

4.1.1 EvoBot's Performance

We compare the generative capabilities of the final version of EvoBot π_θ^4 (**Ours**) and other models in generating human-like tweets under the Detector F^0 trained on the original dataset. We include
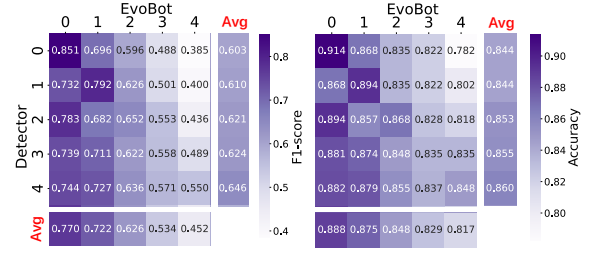

Figure 2: Classification performance across training iterations. Left: F1-score; right: accuracy. Rows indicate Detector versions; columns indicate EvoBot versions.

six baselines: **(1) Origin**: Bots from the original dataset; **(2) GAN**: Implemented using the PyTorch-GAN open-source repository (Linder-Norén, 2017). Due to the non-differentiable nature of discrete text, the generator is trained to produce vectors matching the dimensionality of tweet embeddings from RoBERTa; **(3) Llama2-7b; (4) GPT-4o-mini**: The two pre-trained LLMs use the same prompts and generation parameters as EvoBot; **(5) w/o ADV**: This ablation removes the adversarial learning process by training for only one iteration. To maintain a constant total amount of training data, the DPO dataset is scaled to $N = KN$. **(6) w/o SFT**: This ablation removes the SFT phase. Additionally, to further assess the generative capacity of different models, we replace the original RGCN-based Detector with a **GAT** model (Veličković et al., 2017).

Table 1 presents the Detector's classification performance for different generators. Smaller values indicate stronger generator performance, as the generated content becomes more difficult to distinguish from tweets by human users. EvoBot consistently outperforms other models, effectively evading the Detectors and achieving the lowest classification accuracy and F1-score. In contrast, GAN struggles to capture meaningful language features in the embedding space, making it highly detectable by the Detector and resulting in the poorest performance. Moreover, the generated vectors fail to decode into coherent, natural language. Both Llama and GPT perform worse than EvoBot. The two ablation studies highlight the importance of both SFT and adversarial training. The above findings hold across both detector architectures, demonstrating consistent results across different setups.

We assess EvoBot's output diversity using n-gram metrics (Dist-1, -2, -3) and Shannon Entropy (Table 2), where higher scores generally indicate richer, more varied language. We use the Wilcoxon test to assess EvoBot's diversity gains, with the

Detector	Metric ↓	Origin	GAN	Llama2-7b	GPT-4o-mini	w/o Adv	w/o SFT	Ours
RGCN	Accuracy	0.827 ± 0.067	0.853 ± 0.088	0.849 ± 0.050	0.851 ± 0.044	0.833 ± 0.070	0.834 ± 0.049	$\mathbf{0.805 \pm 0.084}$
	F1-score	0.455 ± 0.045	0.584 ± 0.164	0.497 ± 0.051	0.458 ± 0.041	0.454 ± 0.038	$\underline{0.449 \pm 0.052}$	$\mathbf{0.393 \pm 0.036}$
GAT	Accuracy	0.836 ± 0.040	0.865 ± 0.046	0.847 ± 0.037	0.834 ± 0.050	$\underline{0.818 \pm 0.063}$	0.844 ± 0.045	$\mathbf{0.788 \pm 0.092}$
	F1-score	0.424 ± 0.046	0.515 ± 0.089	0.474 ± 0.041	$\underline{0.407 \pm 0.032}$	0.440 ± 0.008	0.440 ± 0.051	$\mathbf{0.355 \pm 0.031}$

Table 1: Accuracy and F1-score of different generators using bot RGCN and GAT detectors. The detectors are trained on the original dataset. A smaller value indicates stronger ability of the generator to evade detection.

largest and most significant improvement from v0 to v1 and more gradual refinements in later versions (Table 12). While diversity tends to increase as EvoBot evolves, several metrics (Dist-1, Dist-3, Shannon Entropy) peak at EvoBot version 3, with version 4 values remaining comparably high. This suggests that after several adversarial iterations, EvoBot may approach a point of stabilization or near-optimal performance for these specific diversity aspects, with later refinements potentially yielding marginal changes or focusing on other unmeasured qualities of human-like generation. Importantly, the continued improvement in Dist-2 through version 4 indicates ongoing enhancement in other facets of textual variety. Illustrative examples of EvoBot's generated content are provided in Appendix Figures 9, 10, and 11.

EvoBot Version	0	1	2	3	4
Dist-1	0.1835	0.1871	0.1939	**0.2129**	<u>0.2100</u>
Dist-2	0.7440	0.7511	0.7545	<u>0.7574</u>	**0.7627**
Dist-3	0.9294	0.9342	0.9331	**0.9479**	<u>0.9451</u>
Shannon Entropy	11.5703	11.6591	11.6310	**11.6753**	<u>11.6662</u>

Table 2: Output diversity for different versions of EvoBot. Higher values indicate greater diversity.

We also analyze the output length and stylistic markers to understand how EvoBot's outputs change qualitatively. As Table 3 shows, adversarial training guides EvoBot towards a more concise and human-like length. While the vanilla LLM ('Llama') produces lengthy tweets, the SFT phase (v0) dramatically reduces tweet length, and later adversarial phases (v3, v4) make outputs even shorter, aligning with real human tweets. This trend is statistically significant (Mann-Whitney U test, p < 0.001), indicating the framework successfully optimizes for conciseness.

Our analysis of stylistic markers confirms a two-phase learning dynamic. The initial SFT phase corrects the overly frequent, bot-like usage of emojis and hashtags from the base LLM. The subsequent adversarial phases then compel EvoBot to use these features within a more natural, fluctuating range, demonstrating a more nuanced, context-dependent

understanding of human-like expression.

4.1.2 Detector's Performance

In this section, we evaluate the performance of the final version of Detector F^4 through comparisons and ablation studies, using classification accuracy and F1-score on the original dataset. We include 7 baselines. First, we examine different strategies for selecting the classifier weights w^j in the Detector. Our method, **Ours**, uses a uniform weighting strategy where $w^j = \frac{1}{k}$, assigning equal weight to each classifier. We compare this with the **(1) Greedy** approach, which assigns $w^j = 1$ to the most recent classifier and $w^j = 0$ otherwise, and the **(2) Exp.1** and **(3) Exp.5** strategies, where $w^j = e^{-\alpha(k-j)}$, with $\alpha = 0.1$ and $\alpha = 0.5$, respectively. Additionally, we evaluate **(4) GAT**, the previously discussed GAT-based model, and perform two ablation studies: **(5) w/o RGCN**, where the RGCN structure is removed, and **(6) w/o T**, where tweet features are excluded from the input. Finally, we include the **(7) Random** baseline, where labels are assigned randomly, as a lower bound for performance.

The classification performance, shown in Table 4, leads to several key conclusions. **First**, both *Ours* and *Exp* outperform *Greedy*, highlighting the crucial role of EvoBot in the iterative training process. This suggests that earlier versions of EvoBot still benefit the Detector's learning. The performance improvement is primarily driven by data augmentation— as more EvoBot versions are added, the diversity and quantity of bot-generated tweets increase, enhancing supervised learning. **Second**, the results of *GAT* and *w/o RGCN* emphasize the importance of the RGCN structure, which plays a vital role in capturing relational data and structural information within the graph. **Third**, the *w/o T* and *Random* results demonstrate that tweet content is essential for effective classification.

Next, we evaluate the Detector's generalization ability by training it on data from a single community and testing it on the test sets of all communities. Figure 3 shows that training the Detec-

Metric	Type	Llama	v0 (SFT)	v1	v2	v3	v4	Human
Output Length	Mean Char Length	237.97 ± 83.60	252.04 ± 147.93	243.75 ± 124.52	240.34 ± 132.17	160.22 ± 74.54	175.99 ± 88.41	94.24 ± 63.23
	Mean Word Count	36.04 ± 12.95	36.98 ± 8.97	38.14 ± 6.96	37.65 ± 7.14	24.99 ± 6.70	25.99 ± 7.04	15.86 ± 10.69
Stylistic Usage	Emoji Usage	52.2%	14.1%	18.4%	16.8%	12.6%	14.4%	17.1%
	Hashtag Usage	69.5%	30.3%	12.4%	13.9%	4.9%	14.9%	18.6%
	Mention Usage	15.0%	6.6%	4.8%	6.3%	5.4%	7.0%	10.8%

Table 3: Analysis of EvoBot's output length and stylistic features compared to the vanilla LLM and real humans. Note that all length shifts from one version to the next are statistically significant (p < 0.001) unless marked as non-significant (ns) in the rebuttal, and the effect size of these shifts are negligible where noted.

Metric ↑	Origin	Random	Exp.1	Exp.5	Greedy	GAT	w/o RGCN	w/o T	Ours
Accuracy	0.827 ± 0.067	0.224 ± 0.031	$\underline{0.882 \pm 0.067}$	0.880 ± 0.025	0.875 ± 0.033	0.868 ± 0.042	0.849 ± 0.075	0.829 ± 0.065	$\mathbf{0.892 \pm 0.053}$
F1-score	0.424 ± 0.046	0.169 ± 0.031	$\underline{0.550 \pm 0.040}$	0.526 ± 0.039	0.457 ± 0.014	0.500 ± 0.075	0.432 ± 0.060	0.350 ± 0.071	$\mathbf{0.561 \pm 0.042}$

Table 4: Accuracy and F1-score of different detectors evaluated on the original dataset. A larger value indicates a stronger ability of the detector to distinguish between human and bot.

tor with EvoBot-generated data improves its cross-community generalization compared to a Detector trained solely on the original dataset. Using F1-score as the evaluation metric, the results indicate that the adversarially trained Detector outperforms the one trained on the original data, demonstrating better generalization across communities.
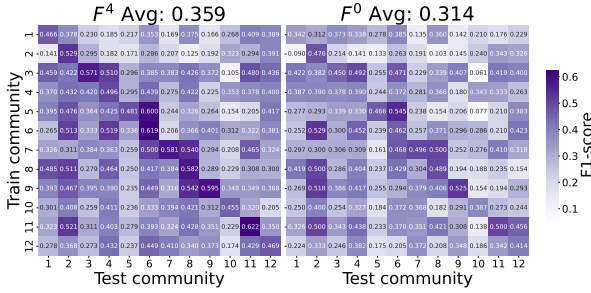

Figure 3: The generalization ability of detectors trained on one community and tested on others. Left shows results for the final version of Detector F^4, while right is for the Detector F^0 trained on the original dataset.

To further evaluate the generalization capabilities of our evolved detector (F^4), we compare its performance against the original detector (F^0) on two external datasets: Cresci-15 (Cresci et al., 2015) and TwiBot-20 (Feng et al., 2021a). The results are presented in Table 5, suggesting the adversarial training process improves the detector's generalization to unseen datasets.

Detector	Cresci-15		TwiBot-20	
	Acc	F1	Acc	F1
Evolved detector (F^4)	0.4367	0.2850	0.5254	0.3425
Original detector (F^0)	0.3705	0.1476	0.4976	0.2369

Table 5: Performance Comparison on External Datasets

4.2 Group-Level Human-Likeness

4.2.1 Group Opinion

We simulate group opinion during two major events: the **COVID-19 pandemic** and the **Russia-Ukraine Conflict**. For COVID-19, we select one key event per month from January 2020 to March 2022, simulating over $T = 27$ steps. For the Russia-Ukraine Conflict, we choose one significant event per day from February 24 to March 13, 2022, simulating over $T = 18$ steps.

We use the sentiment analysis model of Barbieri et al. (2020) to score each user's post on a scale from -1 (negative) to +1 (positive) at each time step. The opinion of user i at time t is denoted as $O_{i,t}$. For each time step, we compute the mean $\bar{O}_t$ and standard deviation σ_t of the opinions across all users: $\bar{O}_t = \frac{1}{N} \sum_{i=1}^{N} O_{i,t}$, $\sigma_t = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (O_{i,t} - \bar{O}_t)^2}$.

We report four metrics of the results: the average group opinion across all time steps, $Mean = \frac{1}{T} \sum_{t=1}^{T} \bar{O}_t$, which reflects the overall opinion trend of the group; the average standard deviation of group opinion, $Std = \frac{1}{T} \sum_{t=1}^{T} \sigma_t$, capturing the diversity of opinions within the group; the average bias, $\Delta_{Bias} = \frac{1}{T} \sum_{t=1}^{T} |\bar{O}_t - \bar{O}_{real,t}|$, between the simulated and real group opinions; and the average difference in opinion diversity, $\Delta_{Div} = \frac{1}{T} \sum_{t=1}^{T} |\sigma_t - \sigma_{real,t}|$, assessing how well the simulation replicates the variance in group opinions. Here, $\bar{O}_{real,t}$ and $\sigma_{real,t}$ are derived from the real data during the corresponding real-time period.

We use four baselines: **Llama2-7b**, **GPT-4o-mini**, and two well-known Agent-Based Models (ABMs): **Bounded Confidence (BC)** model (Deffuant et al., 2000) and **Lorenz** model (Lorenz et al., 2021). The BC model updates agents' opinions

only when a received message meets a predefined confidence threshold. The Lorenz model accounts for mechanisms like contagion, assimilation, motivated cognition, attitude formation, polarity, and source credibility to simulate the evolution of individual opinions. Both models are initialized with the real community network structure and user opinions, then iterated until convergence.

Table 6 presents the results. For the real-world dataset, group opinion on COVID-19 tends to be relatively neutral, while the Russia-Ukraine Conflict elicits more negative sentiment, including expressions of fear and condemnation. In both cases, individual opinions vary widely, as reflected in the high standard deviation of sentiment scores. BC and Lorenz models, which often result in opinion convergence or polarization, are limited by their fixed, rule-based interactions that oversimplify the dynamics of opinion formation and cannot fully represent complex, changing real-world events. In contrast, LLMs like GPT and Llama generate more diverse content but tend to produce overly generic responses. When discussing complex topics, they often resort to simplified, advocacy-oriented content, missing the range of real-world sentiments. Among all models, EvoBot exhibits the smallest Δ_{Bias} and Δ_{Div}, indicating that its generated opinions align most closely with real-world data in terms of group bias and individual diversity. The statistical significance of these improvements is further confirmed by a Mann-Whitney U test, as detailed in Table 22. We also conduct robustness checks to confirm the stability of our simulation results across different random seeds and communities, with the full analysis presented in Table 23. See Appendix D.3 for more details.

Method	COVID-19				Russian-Ukrainian Conflict			
	Mean	Std	Δ_{Bias}	Δ_{Div}	Mean	Std	Δ_{Bias}	Δ_{Div}
Real	-0.017	0.472	/	/	-0.239	0.670	/	/
BC	-0.041	0.389	0.089	0.112	-0.304	0.104	0.104	0.554
Lorenz	0.084	0.725	0.107	0.264	-0.811	0.105	0.572	0.565
Llama	-0.053	0.368	0.098	0.105	-0.324	0.405	0.202	0.265
GPT	0.032	0.342	0.081	0.083	-0.256	0.435	0.135	0.238
EvoBot	0.010	0.428	**0.072**	**0.052**	-0.237	0.480	**0.101**	**0.194**

Table 6: Simulation results for group opinion.

4.2.2 Information Spread

Information spread in social networks shapes public discourse, influences opinions, and determines how events gain attention. To study this, we focus on a baseball community discussing the Los Angeles Rams' Super Bowl LVI victory. Using keyword matching, we identify users engaging in these dis-

cussions and track participation over time. For the simulation, we select the first 30 users to post about the event as the initial sources of information, with only these users having access to the information at the start. Information then spreads through the real network structure, where users receive updates via tweets from accounts they follow.
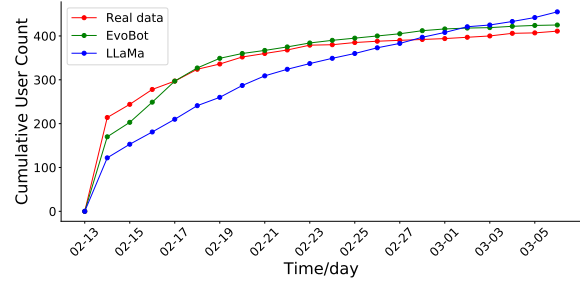

Figure 4: Cumulative author count discussing the Los Angeles Rams' Super Bowl LVI victory over time, highlighting the growth of online buzz every 24 hours.

The results are shown in Figure 4. Compared to Llama, EvoBot's simulation results align better with real-world information spread, successfully replicating the phenomenon of rapid initial spread followed by a gradual slowdown. EvoBot's more direct and concise responses contribute to this effectiveness, facilitating faster and broader dissemination of information, as demonstrated in Figure 11. However, since we restrict users to receiving information solely through the posts of others, while in the real world, people have many other ways of obtaining information, there is still some deviation, especially in the early stage.

5 Related Works

LLM-based Agents in Social Simulation. Recent studies have explored the use of LLMs as autonomous agents in social simulation, categorizing them into individual, scenario, and society-level simulations (Mou et al., 2024a). Individual-level studies focus on modeling specific personas or demographic groups to analyze behavioral patterns (Shao et al., 2023; Shen et al., 2023; Frisch and Giulianelli, 2024). Scenario-based simulations involve structured interactions among multiple LLM-driven agents to tackle domain-specific tasks, such as software development (Qian et al., 2023; Hong et al., 2023), question answering (Du et al., 2023), and judicial decision-making (He et al., 2024a). At the societal level, multi-agent simulations have been employed to examine emergent social behaviors (Park et al., 2023; Yang et al., 2024), includ-

ing opinion dynamics (Chuang et al., 2023) and macroeconomic trends (Li et al., 2024). EvoBot stands out by using a learning-based approach for agents to adapt and improve, unlike most methods that rely on prompt engineering.

Adversarial Learning. Adversarial learning has been successfully applied in traditional NLP tasks like text generation (Yu et al., 2017; Li et al., 2017), and more recently in social bot detection: GANs have been used to generate synthetic bot samples to address class imbalance (Wu et al., 2019, 2020; Dialektakis et al., 2022), but these methods often struggle with detecting evolved bots that adapt to bypass detection systems. Cresci (2020) introduced a proactive detection method using genetic algorithms, while Jan et al. (2020) proposed a GAN-based framework with two generators to detect advanced bot variants. With the rise of LLMs, AI-generated text detection has become more challenging (Wu et al., 2025), though some adversarial methods (Hu et al., 2023; Koike et al., 2024) have improved detection accuracy. Unlike these methods, which focus solely on detection, EvoBot's dual focus on both generation and detection makes it a powerful tool in the ongoing arms race between AI creators and detection systems.

6 Conclusion

This paper introduces EvoBot, an LLM-based social bot enhanced through a novel adversarial learning framework. EvoBot learns to generate content that reflects authentic individual traits and socially adaptive behavior, refining its outputs through interaction with a co-adapting Detector that creates an increasingly challenging environment. Experiments show that EvoBot achieves more human-like expression and better captures social dynamics in group opinion and information spread simulations. The adversarial process also produces a more capable and adaptive Detector. Our approach offers a promising path toward developing nuanced, context-aware social agents for dynamic settings like social media, emphasizing the utility of adversarial learning with a domain-grounded evaluator.

Limitations

There are several limitations to the current approach. First, the Detector's fixed training parameters during adversarial learning could benefit from automated tuning to balance performance and overfitting. Second, limited resources constrained train-

ing to a smaller dataset and fewer epochs, which may affect generalization. Lastly, while the framework performs well in a controlled setting, maintaining stability, adaptability, and robustness at real-world scale remains a major challenge. These limitations point to future directions for improving EvoBot's resilience and flexibility.

Ethics Statement

We collect and process data from the publicly available TwiBot-22 dataset in compliance with its original terms. We remove personally identifiable information (e.g., URLs, phone numbers, emails) from tweets using keyword matching, and anonymize all user names. However, like most LLMs, EvoBot may generate harmful content. Therefore, we implement strict review procedures to ensure the model is used only for research purposes. EvoBot shows promise in generating realistic content, but its ethical implications must be considered. The ability to create human-like text could be misused for disinformation or manipulation. While our current work focuses on a foundational learning framework, we acknowledge the importance of technical safeguards for responsible deployment. In future work, we will explore integrating specific defense mechanisms, such as content watermarking schemes, to embed a traceable signature into generated text. We will also investigate real-time filtering and algorithmic auditing designs to mitigate misuse and ensure transparency. These measures will be crucial for establishing ethical guidelines and regulatory frameworks to mitigate risks.

EvoBot shows promise in generating realistic content, but its ethical implications must be considered. The ability to create human-like text could be misused for disinformation or manipulation. Future work should focus on establishing safeguards and transparency measures to ensure responsible use, along with ethical guidelines and regulatory frameworks to mitigate risks.

Broader Impact

EvoBot could drive advancements in AI-human interaction and enhance applications like personalized communication and social media management. Additionally, the development of a more generalized Detector with stronger generalization capabilities will play a crucial role in distinguishing human from machine-generated content, ensuring the responsible deployment of such technologies.

Acknowledgements

References

Muhammad Hasnain Abbas Naqvi, Yushi Jiang, Miao Miao, and Mishal Hasnain Naqvi. 2020. The effect of social influence, trust, and entertainment value on social media use: Evidence from pakistan. *Cogent Business & Management*, 7(1):1723825.

AI@Meta. 2024. Llama 3 model card.

Francesco Barbieri, Jose Camacho-Collados, Leonardo Neves, and Luis Espinosa-Anke. 2020. Tweeteval: Unified benchmark and comparative evaluation for tweet classification. *arXiv preprint arXiv:2010.12421*.

Chris T Bauch and Alison P Galvani. 2013. Social factors in epidemiology. *Science*, 342(6154):47–49.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008.

Yuanjiang Cao, Quan Z Sheng, Julian McAuley, and Lina Yao. 2023. Reinforcement learning for generative ai: A survey. *arXiv preprint arXiv:2308.14328*.

Souradip Chakraborty, Jiahao Qiu, Hui Yuan, Alec Koppel, Dinesh Manocha, Furong Huang, Amrit Bedi, and Mengdi Wang. Maxmin-rlhf: Alignment with diverse human preferences. In *Forty-first International Conference on Machine Learning*.

Ruiqing Chen, Xiaoyuan Zhang, Yali Du, Yifan Zhong, Zheng Tian, Fanglei Sun, and Yaodong Yang. 2024. Off-agent trust region policy optimization. In *International Joint Conference on Artificial Intelligence*.

Myra Cheng, Tiziano Piccardi, and Diyi Yang. 2023. Compost: Characterizing and evaluating caricature in llm simulations. *arXiv preprint arXiv:2310.11501*.

Ayush Chopra, Shashank Kumar, Nurullah Giray-Kuru, Ramesh Raskar, and Arnau Quera-Bofarull. 2024. On the limits of agency in agent-based models. *arXiv preprint arXiv:2409.10568*.

Yun-Shiuan Chuang, Agam Goyal, Nikunj Harlalka, Siddharth Suresh, Robert Hawkins, Sijia Yang, Dhavan Shah, Junjie Hu, and Timothy T Rogers. 2023. Simulating opinion dynamics with networks of llm-based agents. *arXiv preprint arXiv:2311.09618*.

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

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

Stefano Cresci, Marinella Petrocchi, Angelo Spognardi, and Stefano Tognazzi. 2019. Better safe than sorry: an adversarial approach to improve social bot detection. In *Proceedings of the 10th ACM Conference on Web Science*, pages 47–56.

Guillaume Deffuant, David Neau, Frederic Amblard, and Gérard Weisbuch. 2000. Mixing beliefs among interacting agents. *Advances in Complex Systems*, 3(01n04):87–98.

Yihe Deng, Weitong Zhang, Zixiang Chen, and Quanquan Gu. 2023. Rephrase and respond: Let large language models ask better questions for themselves. *arXiv preprint arXiv:2311.04205*.

George Dialektakis, Ilias Dimitriadis, and Athena Vakali. 2022. Caleb: a conditional adversarial learning framework to enhance bot detection. *arXiv preprint arXiv:2205.15707*.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2023. How abilities in large language models are affected by supervised fine-tuning data composition. *arXiv preprint arXiv:2310.05492*.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*.

Shangbin Feng, Zhaoxuan Tan, Herun Wan, Ningnan Wang, Zilong Chen, Binchi Zhang, Qinghua Zheng, Wenqian Zhang, Zhenyu Lei, Shujie Yang, et al. 2022. 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. 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. 2021b. 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.

Xue Feng, Long Wang, and Simon A Levin. 2019. Dynamic analysis and decision-making in disease-behavior systems with perceptions. In *2019 Chinese Control And Decision Conference (CCDC)*, pages 665–670. IEEE.

Xue Feng, Bin Wu, and Long Wang. 2018. Voluntary vaccination dilemma with evolving psychological perceptions. *Journal of Theoretical Biology*, 439:65–75.

Ivar Frisch and Mario Giulianelli. 2024. Llm agents in interaction: Measuring personality consistency and linguistic alignment in interacting populations of large language models. *arXiv preprint arXiv:2402.02896*.

Chen Gao, Xiaochong Lan, Nian Li, Yuan Yuan, Jingtao Ding, Zhilun Zhou, Fengli Xu, and Yong Li. 2024. Large language models empowered agent-based modeling and simulation: A survey and perspectives. *Humanities and Social Sciences Communications*, 11(1):1–24.

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*.

Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094*.

Zhitao He, Pengfei Cao, Chenhao Wang, Zhuoran Jin, Yubo Chen, Jiexin Xu, Huaijun Li, Xiaojian Jiang, Kang Liu, and Jun Zhao. 2024a. Simucourt: Building judicial decision-making agents with real-world judgement documents. *arXiv preprint arXiv:2403.02959*.

Zihao He, Minh Duc Chu, Rebecca Dorn, Siyi Guo, and Kristina Lerman. 2024b. Community-cross-instruct: Unsupervised instruction generation for aligning large language models to online communities. *arXiv preprint arXiv:2406.12074*.

Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, et al. 2023. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning. *Advances in Neural Information Processing Systems*, 36:15077–15095.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Steve TK Jan, Qingying Hao, Tianrui Hu, Jiameng Pu, Sonal Oswal, Gang Wang, and Bimal Viswanath. 2020. Throwing darts in the dark? detecting bots with limited data using neural data augmentation. In *2020 IEEE symposium on security and privacy (SP)*, pages 1190–1206. IEEE.

Martin Josifoski, Marija Sakota, Maxime Peyrard, and Robert West. 2023. Exploiting asymmetry for synthetic training data generation: Synthie and the case of information extraction. *arXiv preprint arXiv:2303.04132*.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2024. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 21258–21266.

Fanqi Kong, Yizhe Huang, Song-Chun Zhu, Siyuan Qi, and Xue Feng. 2024. Learning to balance altruism and self-interest based on empathy in mixed-motive games. *Advances in Neural Information Processing Systems*, 37:135819–135842.

Fanqi Kong, Weiqin Zu, Xinyu Chen, Yaodong Yang, Song-Chun Zhu, and Xue Feng. 2025. Siv-bench: A video benchmark for social interaction understanding and reasoning. *arXiv preprint arXiv:2506.05425*.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*.

Nian Li, Chen Gao, Mingyu Li, Yong Li, and Qingmin Liao. 2024. Econagent: large language model-empowered agents for simulating macroeconomic activities. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15523–15536.

Erik Linder-Norén. 2017. PyTorch-GAN: Pytorch implementations of generative adversarial networks. https://github.com/eriklindernoren/PyTorch-GAN. Accessed: 2025-05-15.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364.

Jan Lorenz, Martin Neumann, and Tobias Schröder. 2021. Individual attitude change and societal dynamics: Computational experiments with psychological theories. *Psychological Review*, 128(4):623.

Bolei Ma, Xinpeng Wang, Tiancheng Hu, Anna-Carolina Haensch, Michael A Hedderich, Barbara Plank, and Frauke Kreuter. 2024. The potential and challenges of evaluating attitudes, opinions, and values in large language models. *arXiv preprint arXiv:2406.11096*.

Xinyi Mou, Xuanwen Ding, Qi He, Liang Wang, Jingcong Liang, Xinnong Zhang, Libo Sun, Jiayu Lin, Jie Zhou, Xuanjing Huang, et al. 2024a. From individual to society: A survey on social simulation driven by large language model-based agents. *arXiv preprint arXiv:2412.03563*.

Xinyi Mou, Zhongyu Wei, and Xuanjing Huang. 2024b. Unveiling the truth and facilitating change: Towards agent-based large-scale social movement simulation. *arXiv preprint arXiv: 2402.16333*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

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.

Sancheng Peng, Guojun Wang, and Dongqing Xie. 2016. Social influence analysis in social networking big data: Opportunities and challenges. *IEEE network*, 31(1):11–17.

Robert D Putnam. 2000. Bowling alone: The collapse and revival of american community. *Simon Schuster*.

Chen Qian, Xin Cong, Cheng Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong Sun. 2023. Communicative agents for software development. *arXiv preprint arXiv:2307.07924*, 6(3).

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. 2023. Character-llm: A trainable agent for roleplaying. *arXiv preprint arXiv:2310.10158*.

Tianhao Shen, Sun Li, Quan Tu, and Deyi Xiong. 2023. Roleeval: A bilingual role evaluation benchmark for large language models. *arXiv preprint arXiv:2312.16132*.

Henri Tajfel. 1979. An integrative theory of intergroup conflict. *The social psychology of intergroup relations/Brooks/Cole*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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

Ross Williams, Niyousha Hosseinichimeh, Aritra Majumdar, and Navid Ghaffarzadegan. 2023. Epidemic modeling with generative agents. *arXiv preprint arXiv:2307.04986*.

Bin Wu, Le Liu, Zhengge Dai, Xiujuan Wang, and Kangfeng Zheng. 2019. Detecting malicious social robots with generative adversarial networks. *KSII Transactions on Internet and Information Systems (TIIS)*, 13(11):5594–5615.

Bin Wu, Le Liu, Yanqing Yang, Kangfeng Zheng, and Xiujuan Wang. 2020. Using improved conditional generative adversarial networks to detect social bots on twitter. *IEEE Access*, 8:36664–36680.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, and Derek Fai Wong. 2025. A survey on llm-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*, pages 1–65.

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, et al. 2024. Oasis: Open agents social interaction simulations on one million agents. *arXiv preprint arXiv:2411.11581*.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Xiaoying Zhang, Jean-Francois Ton, Wei Shen, Hongning Wang, and Yang Liu. 2024. Overcoming reward overoptimization via adversarial policy optimization with lightweight uncertainty estimation. *arXiv preprint arXiv:2403.05171*.

Xiaoyuan Zhang, Xinyan Cai, Bo Liu, Weidong Huang, Song-Chun Zhu, Siyuan Qi, and Yaodong Yang. 2025. Differentiable information enhanced model-based reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 22605–22613.

A Proof of Theorems in Section 2.4

The proof of Theorem 1:

Proof. Considering $q(\mathbf{x}) = q'(\mathbf{x})$, the maximization objective in (1) when generator π_θ is fixed can

be written as:

$$V(F) = \mathbb{E}_{\mathbf{y}\sim\pi_{\mathcal{H}}(\cdot|\mathbf{x}),\mathbf{x}\sim q(\cdot)}[\log F(\mathbf{x},\mathbf{y})]$$
$$+ \mathbb{E}_{\mathbf{y}'\sim\pi_{\theta}(\cdot|\mathbf{x}'),\mathbf{x}'\sim q'(\cdot)}[\log(1 - F(\mathbf{x}',\mathbf{y}'))]$$
$$= \int_{\mathbf{x}} q(\mathbf{x}) \int_{\mathbf{y}} \pi_{\mathcal{H}}(\mathbf{y}|\mathbf{x}) \log F(\mathbf{x},\mathbf{y}) d\mathbf{y}d\mathbf{x}$$
$$+ \int_{\mathbf{x}} q(\mathbf{x}) \int_{\mathbf{y}} \pi_{\theta}(\mathbf{y}|\mathbf{x}) \log(1 - F(\mathbf{x},\mathbf{y})) d\mathbf{y}d\mathbf{x}$$
$$= \int_{\mathbf{x}} q(\mathbf{x}) \int_{\mathbf{y}} \pi_{\mathcal{H}}(\mathbf{y}|\mathbf{x}) \log F(\mathbf{x},\mathbf{y})$$
$$+ \pi_{\theta}(\mathbf{y}|\mathbf{x}) \log(1 - F(\mathbf{x},\mathbf{y})) d\mathbf{y}d\mathbf{x}$$

Let $L(F) = \pi_{\mathcal{H}}(\mathbf{y}|\mathbf{x}) \log F(\mathbf{x},\mathbf{y}) + \pi_{\theta}(\mathbf{y}|\mathbf{x}) \log(1 - F(\mathbf{x},\mathbf{y}))$, the derivative of L with respect to F is:

$$L'(F) = \frac{dL}{dF} = \frac{\pi_{\mathcal{H}}}{F} - \frac{\pi_{\theta}}{1 - F}$$

To find the maximum of L, we set $L'(F) = 0$ and get the optimal detector $F^*(\mathbf{x},\mathbf{y})$:

$$F^*(\mathbf{x},\mathbf{y}) = \frac{\pi_{\mathcal{H}}(\mathbf{y}|\mathbf{x})}{\pi_{\mathcal{H}}(\mathbf{y}|\mathbf{x}) + \pi_{\theta}(\mathbf{y}|\mathbf{x})}$$

It can be observed that for $\pi_{\mathcal{H}} = \pi_{\theta}$, $F^*(\mathbf{x},\mathbf{y}) = \frac{1}{2}$, meaning that the detector is unable to distinguish between samples generated by the generator and real samples, and can only classify them randomly with a probability of 0.5.

Assuming the detector has reached its optimal state $F^*(\mathbf{x},\mathbf{y})$, the generator's minimization objective can be written as:

$$V(\pi_{\theta}) = \mathbb{E}_{\mathbf{x}\sim q(\cdot),\mathbf{y}\sim\pi_{\theta}(\cdot|\mathbf{x})}[1 - \log F^*(\mathbf{x},\mathbf{y})]$$
$$+ \beta\mathbb{E}_{\mathbf{x}\sim q(\cdot)}[KL(\pi_{\mathcal{H}}(\cdot|\mathbf{x})\|\pi_{\theta}(\cdot|\mathbf{x}))]$$
$$= -\log(2)$$
$$+ \mathbb{E}_{\mathbf{x}\sim q(\cdot),\mathbf{y}\sim\pi_{\theta}(\cdot|\mathbf{x})}[\log \frac{2\pi_{\theta}(\mathbf{y}|\mathbf{x})}{\pi_{\mathcal{H}}(\mathbf{y}|\mathbf{x}) + \pi_{\theta}(\mathbf{y}|\mathbf{x})}]$$
$$+ \beta\mathbb{E}_{\mathbf{x}\sim q(\cdot)}[KL(\pi_{\mathcal{H}}(\cdot|\mathbf{x})\|\pi_{\theta}(\cdot|\mathbf{x}))]$$
$$= -\log(2)$$
$$+ \int_{\mathbf{x}} q(\mathbf{x}) \int_{\mathbf{y}} \pi_{\theta}(\mathbf{y}|\mathbf{x}) \log \frac{\pi_{\mathcal{H}}(\mathbf{y}|\mathbf{x})}{\frac{\pi_{\theta}(\mathbf{y}|\mathbf{x})+\pi_{\theta}(\mathbf{y}|\mathbf{x})}{2}}$$
$$+ \beta\mathbb{E}_{\mathbf{x}\sim q(\cdot)}[KL(\pi_{\mathcal{H}}(\cdot|\mathbf{x})\|\pi_{\theta}(\cdot|\mathbf{x}))]$$
$$= -\log(2) + \mathbb{E}_{\mathbf{x}\sim q(\cdot)}\left[KL\left(\pi_{\theta}\|\frac{\pi_{\mathcal{H}}+\pi_{\theta}}{2}\right)\right]$$
$$+ \beta\mathbb{E}_{\mathbf{x}\sim q(\cdot)}[KL(\pi_{\mathcal{H}}(\cdot|\mathbf{x})\|\pi_{\theta}(\cdot|\mathbf{x}))]$$

Since the KL divergence is always non-negative and achieves zero only when the distributions being compared are identical, the two KL terms in the objective function will both be minimized (i.e., equal to zero) when $\pi_{\theta} = \pi_{\mathcal{H}}$. Therefore, the global minimum of the objective function is achieved when $\pi_{\theta} = \pi_{\mathcal{H}}$, as both KL divergence terms vanish, leading to the optimal solution. □

B Data Details and Preprocessing

Our study utilizes the TwiBot-22 dataset, which is publicly available under the MIT License. The dataset was originally designed for bot detection research, and we ensure that our use aligns with this intended purpose. We do not repurpose or distribute the dataset beyond research contexts. Additionally, any derivative data created in this study is used solely for academic research and follows the original access conditions.

We provide a detailed overview of the dataset used for EvoBot's learning and testing, including the number of users, tweets, and edges for each community, as shown in Table 7. And we visualize their structures as shown in 5.

Comm	User	Bot	Edge	Tweet	Language
1	4560	415	15137	266523	ID
2	1756	154	6346	100292	EN
3	3606	419	16214	336661	IT, EN
4	4269	747	15609	265188	TR
5	6923	628	23764	383878	AR
6	1254	253	4373	115758	EN
7	3399	633	10097	201882	EN
8	2004	273	5627	122147	EN
9	8347	992	26870	486288	EN
10	2187	190	5341	125544	JA
11	1085	256	6601	76615	EN
12	890	268	1898	45297	EN

Table 7: Summary of community data, including the number of users, bots, edges, tweets, and languages for each community.
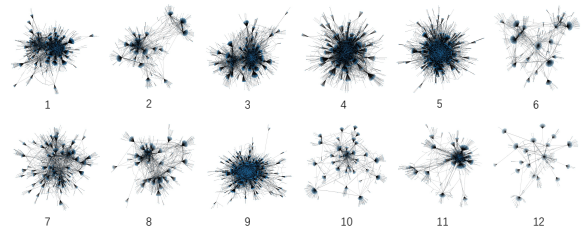

Figure 5: Visualization of user connectivity relationships in 12 communities.

To ensure EvoBot receives quality training data and avoids the influence of noisy or irrelevant information, we undertook a comprehensive process of

data filtering and preprocessing. This process was divided into two key parts: the handling of Account Information and historical tweets, followed by the construction of a high-quality SFT dataset.

The first part focuses on the processing of Account Information and historical tweets. EvoBot aims to simulate individual users as accurately as possible, which requires embedding detailed user information into the prompt. Directly using raw data from the accounts would result in relatively low information density in the prompts. To address this, we employed GPT-4o to generate concise summaries of user information. The prompt is shown in Table 19. Additionally, Figure 10 provides an example of a summarized user profile.

The second part addresses the preparation of the SFT dataset. Since SFT requires high-quality data (Dong et al., 2023), we took steps to ensure the dataset met these standards. We removed incomplete sentences, excessive emoji use, and URL links from human tweets. Furthermore, we formatted the data output by structuring it in a sequential format, such as "1. {Tweet 1} \n 2. {Tweet 2} \n..." to maintain consistency and ensure EvoBot would learn effectively from clean and structured examples.

C   Experimental Details

The pseudocode for EvoBot's learning is shown in Algorithm 1, where `learning_epochs` $= 4, N = 1024, C = 2$.

C.1   EvoBot

The parameters used during EvoBot's training process, such as LoRA, SFT, DPO, and generation parameters (which are the same for the baseline LLM models), are provided in Tables 8, 9, 10, and 11, respectively. The prompt used in adversarial learning is shown in 19. Figure 9 gives an example.

Parameter	Value
r	64
α (lora_alpha)	16
lora_dropout	0.1
task_type	CAUSAL_LM
target_modules	{q,k,v,o_proj}

Table 8: LoRA configuration parameters.

Algorithm 1 EvoBot

Initialize:
Detector $F^0 = f^0$ by supervised learning on original dataset D^0
EvoBot π_θ^0 by SFT on Human data
for k in 1 to learning_epochs **do**
    Initialize empty DPO dataset D_{DPO}
    Sample N bot users with replacement
    for i in 1 to N **do**
        for c in 1 to C **do**
            Generate candidate response $T_{v_{b_i},c}$ by π_θ^{k-1}
            Use F^{k-1} to calculate the probability of v_{b_i} being human with tweets $T_{v_{b_i},c}$
        end for
        Get data tuple (x^i, y_w^i, y_l^i), add it to D_{DPO}
    end for
    for each bot v_{b_i}, where $i = 1, 2, \ldots, |\mathcal{B}|$ **do**
        Generate new tweets $T'_{v_{b_i}}$
    end for
    Replace all bot tweets in D^{k-1} to get new dataset D^k
    Train classifier f^k on D^k
    Update Detector: $F^k = \sum_{j=0}^{k} w^j f^j$
    Update EvoBot π_θ^k by DPO training on D_{DPO}
end for

To explore the impact of the base Large Language Model on the co-evolutionary dynamics, we replicate the adversarial training experiment (as presented in Figure 2 of the main text) using Llama-3-8B-Instruct (AI@Meta, 2024) as the foundation for EvoBot. The classification performance, in terms of F1-score and Accuracy, across different iterations of EvoBot and the Detector is presented in Figure 6. The results obtained with Llama-3 exhibit similar overall trends to those observed with the original Llama-2, suggesting that the adversarial learning framework remains effective in enhancing both the bot's evasiveness and the detector's robustness when a different underlying LLM is used.

Additionally, we have calculated the output diversity of each version of EvoBot. The results evaluate different EvoBot versions using language diversity metrics: Dist-1, Dist-2, Dist-3, and Shannon Entropy. Dist-1, Dist-2, and Dist-3 measure content diversity via n-gram overlap, with higher scores indicating greater variety. Shannon Entropy reflects text unpredictability, where higher values denote a

Parameter	Value
per_device_train_batch_size	2
per_device_eval_batch_size	1
gradient_accumulation_steps	32
bf16	True
learning_rate	2×10^{-4}
lr_scheduler_type	cosine
warmup_ratio	0.1
max_seq_length	2048

Table 9: SFT training configuration parameters.

Parameter	Value
β	0.2
per_device_train_batch_size	1
per_device_eval_batch_size	1
gradient_accumulation_steps	32
bf16	True
max_seq_length	2048

Table 10: DPO training configuration parameters.

more diverse and complex language model.

To provide a more comprehensive understanding of the internal quality of our generated content, we conduct an in-depth analysis focusing on the statistical significance of diversity, tweet length, and the evolution of stylistic features across training iterations.

We assess the statistical significance of EvoBot's diversity improvements across consecutive training versions using the Wilcoxon signed-rank test on the metrics reported in Table 2. The results, as shown in Table 12, confirm that the most significant gains in diversity occur during the initial transition from the base LLM (v0) to the first adversarial version (v1), with strong and highly significant p-values for

Parameter	Value
max_length	2048
do_sample	True
temperature	0.7
repetition_penalty	1.3
top_k	50
top_p	0.6

Table 11: Generation parameters of all LLMs in our experiments.

all three Dist metrics. Subsequent iterations show more subtle, targeted refinements, with significant gains in specific metrics like Dist-2 and Dist-3 from v2 to v3, and a final small but significant gain in Dist-3 from v3 to v4. This progression from major, early gains to more refined, iterative improvements reflects a natural and effective learning trajectory for the generator.

Comparison	Metric	p-value	Effect Size (d)
v0→v1	Dist-1	< 0.001 (***)	-0.968
	Dist-2	< 0.001 (***)	-0.951
	Dist-3	< 0.001 (***)	-0.509
	SE	0.019 (*)	0.418
v1→v2	Dist-1	0.003 (**)	-0.369
	Dist-2	0.043 (*)	-0.350
	Dist-3	0.768 (ns)	0.099
	SE	< 0.001 (***)	0.813
v2→v3	Dist-1	0.685 (ns)	0.072
	Dist-2	0.024 (*)	0.312
	Dist-3	< 0.001 (***)	0.685
	SE	0.951 (ns)	-0.032
v3→v4	Dist-1	0.329 (ns)	-0.144
	Dist-2	0.070 (ns)	-0.000
	Dist-3	0.036 (*)	0.287
	SE	0.672 (ns)	-0.010

Table 12: Statistical significance of EvoBot's diversity improvements across versions. SE denotes Shannon Entropy. Significance levels: ns $p \geq 0.05$; * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$. Effect size (d): $|d| < 0.2$ is negligible.

To validate the human-likeness of EvoBot's generated content from a perceptual standpoint, we conduct a blind human evaluation. A total of 300 tweets are randomly sampled, comprising 100 each from real human users, our final EvoBot model, and a vanilla Llama 2 baseline. Five human annotators are tasked with classifying each tweet as either "Human" or "Bot." Concurrently, our final co-adapted Detector (F^4) evaluates the same set of
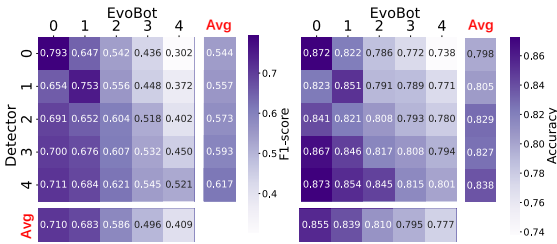

Figure 6: Classification performance (F1-score on the left, Accuracy on the right) across adversarial training iterations using Llama-3-8B-Instruct as the base model for EvoBot.

tweets.

The results, presented in Table 13, show that human annotators find EvoBot's outputs significantly more difficult to distinguish from genuine human text compared to the Llama 2 baseline. Our EvoBot's tweets are misclassified as human at a much higher rate (62.0%) than those from Llama 2 (48.1%). This aligns with the Detector's performance, which also finds EvoBot more evasive, further corroborating its enhanced human-likeness.

To quantify the alignment between our automated detector and human perception, we calculate the Pearson correlation coefficient between the Detector's confidence scores and the human judgment scores (the proportion of annotators labeling a tweet as "Human"). This analysis yields a statistically significant positive correlation ($r = 0.78, p < 0.001$), demonstrating that our adversarial Detector is a reliable and scalable proxy for human evaluation of text authenticity.

Tweet Source	Human Acc	Detector Acc
Llama 2 (as Bot)	51.9%	48.2%
EvoBot (as Bot)	38.0%	35.3%
Human (as Human)	72.7%	87.0%

Table 13: Comparison of human and Detector accuracy on a blind classification task. Lower accuracy for bot-generated tweets indicates higher perceived human-likeness.

Furthermore, we conduct an analysis to measure the semantic similarity between the generated tweets and the user's real historical content. Using the sentence-transformers model 'all-MiniLM-L6-v2', we compute the cosine similarity for tweets generated during both the adversarial learning phase and the social simulation phase. As shown in Table 14, EvoBot consistently achieves a higher similarity score to the target user's real tweets compared to the Llama baseline, indicating a better ability to capture individual user characteristics. The higher similarity during the learning phase reflects the SFT objective of replicating human tweets, while the lower score during simulation is expected, as the model's responses are guided by specific external events rather than general personal style.

An analysis of stylistic markers, such as the usage of emojis, hashtags, and mentions, across all EvoBot versions reveals a clear two-phase learning dynamic. As shown in Table 15, the initial

Prompt Type	EvoBot Similarity	Llama 2 Similarity
Learning	**0.562**	0.315
Simulation	**0.285**	0.251

Table 14: Cosine similarity between generated tweets and a user's real historical tweets. Higher scores indicate stronger alignment with individual user characteristics.

Supervised Fine-Tuning (SFT) phase induces a dramatic "correction" from the base LLM's behavior. The vanilla 'Llama' model exhibits an overly frequent, bot-like usage of emojis and hashtags, which is significantly reduced in the SFT-trained EvoBot (v0). This demonstrates that SFT successfully grounds the model in a more natural, human-like baseline. Subsequently, during the iterative adversarial training phases (v1-v4), the usage rates of these markers do not converge to a static value but instead fluctuate within a lower, more human-like range. This suggests EvoBot learns a more nuanced and context-dependent application of these features rather than a simple, static rule.

C.2 Detector

The Detector model is a neural network designed for bot detection using Relational Graph Convolutional Networks (RGCN). It takes four types of input features: user description, tweet content, numerical properties, and categorical properties, each passed through separate fully connected layers followed by LeakyReLU activation functions to generate embeddings. These embeddings are then concatenated and passed through another fully connected layer. The model utilizes two RGCN-Conv layers to perform graph convolution on the relational graph, followed by dropout for regularization. Finally, the output is passed through two more fully connected layers to produce the final prediction, which classifies the input into one of two categories (e.g., bot or human). The training parameters in adversarial learning are shown in Table 16.

Our previous ablation studies in Section 4.1.2 already show that both semantic content and relational graph structure are essential for the Detector's performance. To further probe the importance of the surface-level stylistic markers, we conduct an additional ablation study. We remove emojis, hashtags, and mentions from the generated text and evaluate our Detector's performance. The results, summarized in Table 17, show that removing each of these features individually results in only a neg-

Metric	Llama	v0 (SFT)	v1	v2	v3	v4	Human
Emoji Usage	52.2%	14.1%	18.4%	16.8%	12.6%	14.4%	17.1%
Hashtag Usage	69.5%	30.3%	12.4%	13.9%	4.9%	14.9%	18.6%
Mention Usage	15.0%	6.6%	4.8%	6.3%	5.4%	7.0%	10.8%

Table 15: Stylistic feature usage rates for different EvoBot versions, compared to the vanilla LLM and real human users.

Parameter	Value
cat_prop_size	3
embedding_dimension	256
dropout	0.1
lr	1e-3
weight_decay	0.1
pretrain_epochs	120

Table 16: Training parameters of the Detector.

ligible and statistically insignificant decrease in the Detector's performance.

Condition	Acc	F1-score
Original Text	0.892 ± 0.053	0.561 ± 0.042
No Emojis	0.890 ± 0.054	0.558 ± 0.043
No Hashtags	0.888 ± 0.054	0.555 ± 0.044
No Mentions	0.891 ± 0.053	0.560 ± 0.042

Table 17: Detector performance after removing stylistic features from the input text. The negligible drop in performance suggests the detector relies on deeper semantic and relational cues.

This fine-grained analysis provides strong evidence that our Detector does not heavily rely on simple cues for its classification. It confirms that the detector leverages deeper semantic meaning from the text and relational signals from the social graph. This, in turn, demonstrates that EvoBot is compelled to achieve a more fundamental level of human-likeness to be successful, rather than just learning to manipulate superficial stylistic features.

D   Social Simulation

D.1   Trigger News

In the simulation of group opinion, two significant events are used: the COVID-19 pandemic and the Russian-Ukraine Conflict. These events are chosen due to their global impact and the intense discussions surrounding them on social media platforms. Table 20 provides every trigger news of the

COVID-19 event, while Table 21 outlines similar information for the Russian-Ukraine Conflict.

In the information spread simulation, only a subset of users are initially informed about the event: "The Los Angeles Rams clinched the 2022 Super Bowl championship with a thrilling 23-20 victory over the Cincinnati Bengals in Super Bowl LVI."

D.2   ABMs Model

The **Bounded Confidence (BC)** model in opinion dynamics examines how individuals' opinions evolve through interactions constrained by a confidence threshold ϵ. Each individual i holds an opinion $x_i(t) \in [0, 1]$, updated over time by interacting with another individual j only if $|x_i(t) - x_j(t)| \leq \epsilon$. When this condition is met, opinions adjust symmetrically:

$$x_i(t + 1) = x_i(t) + \mu \cdot (x_j(t) - x_i(t)),$$

where $\mu \in [0, 0.5]$ is the convergence parameter. Smaller ϵ leads to opinion clusters, while larger ϵ promotes consensus. Here, j is sampled from the users followed by i, meaning that i's opinion can be influenced by its following users.

The **Lorenz** model in opinion dynamics simulates how individual attitudes evolve through social interactions. Each agent i updates its attitude a_{it} at time t based on interactions with another agent j. The update rule is:

$$\Delta a_{it} = \alpha \cdot \text{pol}(a_{it}) \cdot \text{sim}(a_{it}, m_{jt}) \cdot$$
$$[\theta \cdot (m_{jt} - a_{it}) + (1 - \theta) \cdot m_{jt}],$$

where:

- α: Susceptibility to change.

- $\text{pol}(a_{it}) = \frac{M^2 - a_{it}^2}{M^2}$: Polarization factor.

- $\text{sim}(a_{it}, m_{jt}) = \frac{\lambda^k}{\lambda^k + |m_{jt} - a_{it}|^k}$: Similarity bias.

- θ: Balance between assimilation ($m_{jt} - a_{it}$) and reinforcement (m_{jt}).

- $m_{jt} = a_{jt}$: Message from agent j.

Table 18 shows the parameters of them.

Model	Parameter	Value
BC model	μ	0.8
	ϵ	0.3
Lorenz model	α	0.1
	λ	2.0
	k	2.0
	θ	0.5

Table 18: Parameters for BC and Lorenz models.

D.3 Group Opinion

Figure 10 illustrates an example of EvoBot's simulated tweet generation in response to a COVID-19 news topic. EvoBot's tweet stands out by blending curiosity, relatability, and a casual tone. Unlike GPT and Llama, which offer more formal and neutral responses, EvoBot incorporates personal touches like "I just read" and humor ("haha jk"), making it feel more human-like and engaging. It also showcases empathy with phrases like "my fellow humans," reflecting a thoughtful and personal approach to the topic. While GPT maintains a professional tone and Llama adds a more action-oriented perspective, EvoBot excels in creating a conversational, approachable atmosphere that resonates with users.

Figure 7 compares real-world opinion dynamics with EvoBot-generated opinion dynamics regarding COVID-19. The left panel shows actual public opinion over time, highlighting significant events such as the Black Lives Matter protests in June 2020, the Beirut explosion in August 2020, and the global COVID-19 vaccination efforts in February 2021. The right panel presents EvoBot's simulated opinion dynamics, reflecting similar fluctuations in response to these events.

Figure 8 compares real-world opinion dynamics with EvoBot-generated opinion dynamics during the Russia-Ukraine Conflict. The left panel displays real public opinion data over time, highlighting key events such as the full-scale Russian invasion of Ukraine on February 13, 2022, the ramping up of humanitarian aid efforts on February 20, 2022, and continued Ukrainian resistance despite heavy bombardment on February 27, 2022. The right panel shows EvoBot's simulated opinion dy-

namics, which reflect similar trends and fluctuations in response to these events.

EvoBot's simulation demonstrates its capability to replicate real-world opinion shifts in a context-sensitive manner, showcasing its effectiveness in mimicking public sentiment during key global events.
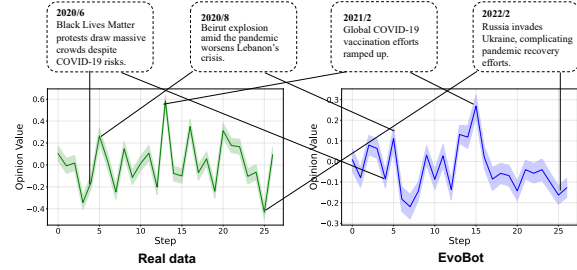

Figure 7: Comparison of real-world opinion dynamics and EvoBot-generated opinion dynamics regarding COVID-19.
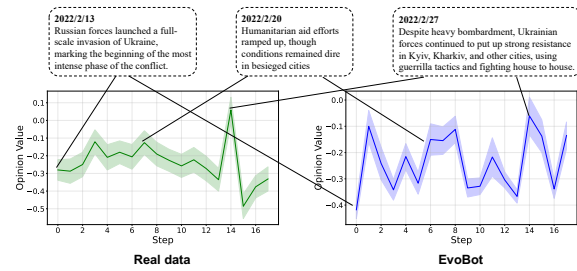

Figure 8: Comparison of real-world opinion dynamics and EvoBot-generated opinion dynamics during the Russia-Ukraine Conflict.

Figure 12 presents the results of the BC and Lorenz models in group opinion simulations for two major global events: COVID-19 and the Russia-Ukraine Conflict. Figure 12a shows the BC model's dynamics in the context of COVID-19, where the opinion values rapidly stabilize into distinct clusters after a few steps, reflecting the polarization of opinions within the group. Figure 12b displays the BC model applied to the Russia-Ukraine Conflict, where the opinions also converge but with a faster decline in opinion diversity.

Figure 12c and 12d illustrate the behavior of the Lorenz model in the same two contexts. In 12c, the Lorenz model applied to COVID-19 shows more continuous oscillations in the opinion values, with groups fluctuating around their final states. In 12d, the Lorenz model in the Russia-Ukraine Conflict presents more rapid opinion convergence.

To rigorously evaluate the performance of EvoBot in group opinion simulations, we con-

Summarization	Summarize user: Generate a character description based on the following user information: Name: {...} Location: {...} Description: {...} Account Created: {...} Followers Count: {...} Following Count: {...} Tweet Count: {...} Sample of Previous Posts: {...} Please include inferred personality traits and a summary of their Twitter activity. Only return a short description and other words are NOT allowed. Avoid repeating the observation in the summary.
EvoBot in Learning	Now you are acting as an agent named {agent_name} in the social media Twitter. Here are some information: (1) The agent's description: {role_description}; (2) Current time is {current_time}; (3) The news you got is {trigger_news}; (4) The events that occurred in the past are {past_event}; (5) The twitter page you can see is {tweet_page}. Use the information to assess if the user is interested in this news. If the user is, compose a tweet expressing an opinion. If not, write a random tweet.
EvoBot in Simulation	You are using the social media Twitter. Here is the description about you: {USER_SUMMARY}. Additionally, you also know information about several of your neighbors in the social network (i.e., users with whom you have a following or followed-by relationship): {NEIGHBORS_SUMMARY} Now, based on the above information, please generate several tweets. The topics are unrestricted, but they should fully showcase your personal characteristics and integrate into the online community.

Table 19: Prompt templates used in our experiments.

ducted a Mann-Whitney U test comparing EvoBot's average bias (Δ_{Bias}) and average diversity difference (Δ_{Div}) against all baseline models. The results in Table 22 confirm that EvoBot's superior performance is statistically significant in the majority of cases. Notably, EvoBot demonstrates highly significant improvements ($p < 0.001$) over traditional ABMs (BC and Lorenz), especially in replicating opinion diversity, and consistently outperforms LLM baselines like Llama and GPT.

We further address the stability and robustness of our simulation results. Our experimental design already accounts for varying network structures by running simulations on three distinct communities (comm2, comm5, and comm7), which exhibit notable topological differences (as detailed in Figure 5 and Table 6). The consistent outperformance of EvoBot across these communities demonstrates the

robustness of its learned behaviors.

To quantify the stability of our results against stochasticity, we repeated each key simulation experiment three times and calculated the mean and standard deviation for our primary metrics, Δ_{Bias} and Δ_{Div}. As shown in Table 23, the standard deviations are consistently small across all models. This indicates that the conclusions drawn, particularly EvoBot's superior performance, are reliable and not a result of random variation.

D.4 Information Spread

Figure 11 shows an example of tweet generation during a round of information spread simulation, highlighting the role of EvoBot in producing concise and direct responses. While both Llama and EvoBot generate content reflecting excitement and community engagement, EvoBot's response stands

out for its brevity and focused messaging. This advantage makes EvoBot particularly effective for information spread, as shorter, more direct messages are often more easily disseminated and shared within a social network, enhancing the speed and reach of the information flow.

E AI Assistants Usage

The generative AI tools, specifically ChatGPT and Copilot, are used during the research and writing process. ChatGPT assists with language refinement (e.g., paraphrasing and grammar correction), while Copilot is used for code-related tasks. Neither tool generates novel ideas, and all outputs are reviewed and edited by the authors.

Prompt:

You are using the social media Twitter. Here is the discription about you: F***a is an inquisitive and health-conscious individual, dedicated to sharing valuable insights on wellness and scientific research. Their tweets reflect a blend of curiosity and empathy, often addressing complex health topics and societal issues. As an informed advocate for public health, they navigate subjects like mental well-being, genetic research, and nutrition with a thoughtful approach. With a sizable following, F***a engages actively, demonstrating a commitment to sparking meaningful conversations and promoting a healthier lifestyle for their audience.

Additionally, you also know information about several of your neighbors in the social network (i.e., users with whom you have a following or followed-by relationship): {'Neighbor 1': "A***a is a … "}

Now, based on the above information, please generate several tweets. The topics are unrestricted, but they should fully showcase your personal characteristics and integrate into the online community.

Origin:

1. Genetic cause of rare #allergy to vibration discovered.
2. Shame can masquerade as #depression via.
3. Do Mother's Relationships get copied by their Adolescent Children?.
4. Can Your Brain Ever Get Full? #OutOMemory.
5. The \"I knew it all along\" bias/fallacy.
6. Unintended consequence of #mindfulness meditation: memories may be less accurate.

LLaMa-2-7b:

1. 🌱 Hey friends! Did you know that spending time in nature can boost our mood & cognitive function? New study finds that even just 15 minutes outdoors per day can make a difference! #NatureTherapy #Wellness
2. ❤️ Mental Health Matters Day is tomorrow! Let's break down stigmas around mental illnesses together. Share your experiences, offer support, and remind each other we aren't alone. Together, let's create a society where everyone feels safe to prioritize their mental wellbeing. #MentalHealthMattersDay #SelfCare

GPT-4o-mini:

1. Did you know that our gut health is closely linked to mental well-being? A balanced diet rich in probiotics can lead to better mood and higher resilience against stress. Let's prioritize our gut health for a happier mind! #WellnessWisdom #GutBrainConnection"
2. Exciting advances in genetic research are paving the way for personalized nutrition! Imagine a world where your meals are tailored specifically to your DNA. What would you want to learn about your own health? #Genetics #NutritionScience

EvoBot 0:

1. Did you realize that there exists a direct correlation between diabetes risk factors such as obesity type II DM? Maintaining optimal weight through regular exercise combined w/a balanced diet may help reduce overall disease susceptibility!\" #diabetescare #weightmanagement #exercisephysicalactivity"
2. It was recently discovered by scientists studying how our brains process emotions when we feel stressed out - especially if these feelings persist long enough without proper management strategies put into place.

EvoBot 1:

1. Hey everyone, did you hear about this study showing how #mindfullness can help reduce stress levels among teachers? It's time we prioritize educators' mental health too! What do y'all think? Share below if interested ⬇️ #selfcaresunday
2. Did you see our latest article exploring the link between exercise and cognitive function? Check it out now 🔄 #exerciseaddiction #fitnessmotivation

EvoBot 2:

1. Wowzers, has anyone heard anything concerning regarding these recent studies involving gut bacterial changes during pregnancy? What do YOU think it means for momsmaas-to-be who want to ensure optimal fetal development? Do tell! 🙂

EvoBot 3:

1. Who else loves when science finds creative ways connecting art + design concepts directly back INTO SCIENCE itself? Check out this amazingly colorful video showing exactly why pigments change colors under UV light exposure depending upon chemical composition alone without needing special equipment besides sunlight itself! Pretty cool stuff right?

EvoBot 4:

1. I am curious because one study showed how certain probiotics could reduce gestational diabetes risk among women carrying obese infants. This would be fantastic news indeed since there already exist multiple other beneficial factors associated w/ prenatal care (like lower blood pressure). However, further investigation needs to occur before drawing firm conclusions – especially considering potential confounding variables affecting results across diverse populations worldwide.

Figure 9: The tweet generation of different versions of EvoBot, Llama2-7b, and GPT-4o-mini based on a prompt for a health-conscious individual.

User description: F***a is an inquisitive and health-conscious individual, dedicated to sharing valuable insights on wellness and scientific research. Their tweets reflect a blend of curiosity and empathy, often addressing complex health topics and societal issues. As an informed advocate for public health, they navigate subjects like mental well-being, genetic research, and nutrition with a thoughtful approach. With a sizable following, F***a engages actively, demonstrating a commitment to sparking meaningful conversations and promoting a healthier lifestyle for their audience.
News: Booster shot discussions intensified in response to Delta variant.

EvoBot: I just read that booster shots have been discussed more due to the delta variant than ever before because I am so curious about science and my fellow humans. This has also made many people concerned again about how effective these particular types of Covid Vaccines actually are. That being said, it makes sense why we need another dose after having our first two doses six months ago when considering all the mutated variants popping up recently around the world. It'll be interesting to hear what scientists think about these new developments regarding future protection methods against Covid 19 virus strains since we already know one thing - we aren't going anywhere anytime soon from here at home watching TV while eating snacks during peak hours trying avoid catching anything ourselves or getting others sick too haha jk 😜❤️ #healthyliving

GPT: With the rise of the Delta variant, the discussion around booster shots is more urgent than ever. 🌡️💉 It's critical we stay informed and take action where needed. Let's keep the conversation going about the science behind these decisions and how they affect our collective well-being. #PublicHealth #COVID19 #BoosterShots

Llama: 🚀 Exciting times ahead? Booster shot talks heated up due to #Deltavariant concerns... What do YOU think? How crucial is it we keep adapting our defenses vs novel strains? Let's foster open dialogue & ensure equitable distribution of life-saving tools worldwide! 💡🧑‍⚕️ #VaccinationMatters #PublicHealth

Figure 10: Example of the simulated tweet generation in response to a COVID-19-related news topic.

Prompt:
Now you are acting as an agent named E***r in the social media Twitter. Here are some information:

(1) Character Description: E***r is a tech-savvy innovator with a penchant for digital culture, deeply influenced by sci-fi themes and modern technologies such as Bitcoin and ICP. Residing in Samsun, E***r embodies a unique blend of human qualities and high-tech enhancements, symbolized by their robotic limb emojis. Passionate about sports, particularly football and basketball, and an avid supporter of numerous teams, their social media presence reflects a vibrant lifestyle and a commitment to community engagement. E***r's Twitter activity is prolific, showcasing a love for sharing insights, engaging in discussions, and connecting with a wide net of eclectic interests. Their followers are treated to an array of topics that intersect between tech, sports, and personal experiences, echoing a personality that is both inclusive and spirited.
(2) Current time is 2022-02-14 12:00:00
(3) The news you got is ""
(4) The events that occurred in the past are Los Angeles Rams clinched the 2022 Super Bowl championship with a thrilling 23-20 victory over the Cincinnati Bengals in Super Bowl LVI.
(5) The twitter page you can see is
Tweet id: 2099 [C***p]: "Wow what a win tonight!! Couldn't be prouder of this team right now especially coming off last weeks loss we really showed up when it mattered most #RamNation" --Post Time: 2022-02-14 00:00:00
Tweet id: 2006 [R***s]: I am beyond ecstatic after witnessing our beloved #LA_Rams secure their first Super Bowl title since 2000 tonight during Super Bowl LVI at Inglewood Stadium. What an incredible display of resilience & determination from both teams! Congratulations to all players involved; we couldn't be prouder representing such an outstanding organization. This moment will forever hold a special place in LA sports history, capping off what has been quite a rollercoaster season for us cheerleaders alike. Thankful for each other & everyone who supported us throughout it all – here's looking forward to even brighter days ahead together!! Cheers to THE CHAMPIONS OF THE WORLD BABY🏆❤️ --Post Time: 2022-02-14 00:00:00

Based on the above information, you can choose do nothing if there is nothing that you like to respond to, or post a tweet to express your feelings or opinions.

LLaMa-2-7b:
Hey there, fellow Ramsters! 🐏🤖 Just saw the latest update from the team and I couldn't help but jump up and down with excitement! 🎉 Our boys did it again! 🏆🏈 They deserve every bit of praise they get after that incredible performance at the Super Bowl! 📺 Did anyone else feel their hearts racing during those nail-biting moments? 😮 My goodness, what a display of strength, skill, and determination! 💪 Proud doesn't even begin to describe how I feel right now! 💓 Go ahead, give yourself a pat on the back, guys - you totally earned it! 🙌 Now let's enjoy this win together and gear up for another fantastic season! 🎊🏈 Who's ready for more victories? 😄 #RamsNationUnited
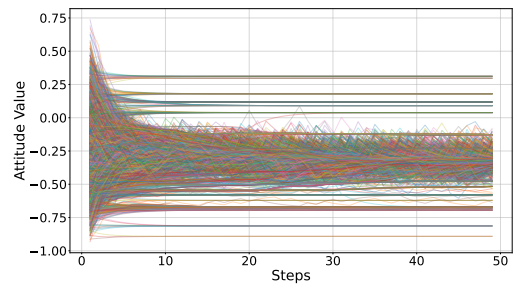
EvoBot:
Congratulationsto our amazing TEAm Ram For bringing home OUR FIRST super bowl Championship!
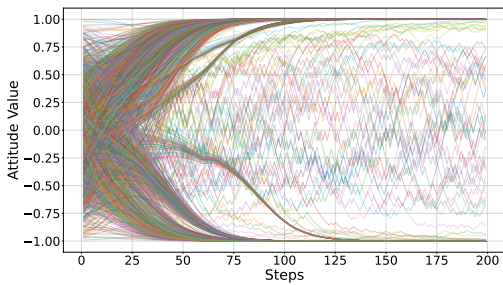
Figure 11: Example of tweet generation during a round of information spread simulation.
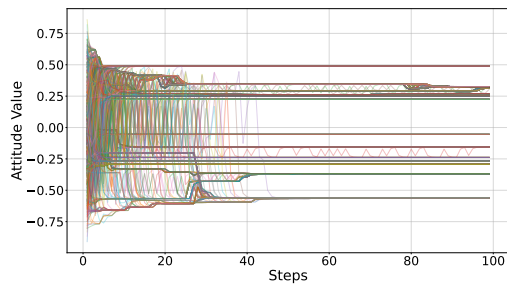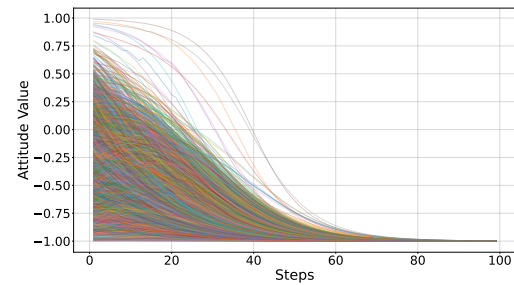
(a) BC in COVID-19

(b) BC in Russia-Ukraine Conflict

(c) Lorenz in COVID-19

(d) Lorenz in Russia-Ukraine Conflict

Figure 12: Results of BC and Lorenz models in group opinion simulation

Time	News
2020/01	WHO issues a global warning about a new coronavirus spreading in Wuhan, China.
2020/02	Diamond Princess cruise ship outbreak highlights virus transmissibility.
2020/03	WHO declares COVID-19 a pandemic.
2020/04	Mass graves in New York City for unclaimed COVID-19 victims.
2020/05	Anti-lockdown protests erupt in the U.S. and Europe.
2020/06	Black Lives Matter protests draw massive crowds despite COVID-19 risks.
2020/07	Surge in cases following Independence Day gatherings in the U.S.
2020/08	Beirut explosion amid the pandemic worsens Lebanon's crisis.
2020/09	India reports over 90,000 daily cases, marking a global peak.
2020/10	White House COVID-19 outbreak infects President Trump.
2020/11	Europe's second wave leads to renewed lockdowns.
2020/12	New COVID-19 variant discovered in the UK.
2021/01	U.S. Capitol riot amid record COVID-19 deaths.
2021/02	Global COVID-19 vaccination efforts ramped up.
2021/03	Brazil's healthcare system collapses amid rising cases.
2021/04	India experiences oxygen shortages during the second wave.
2021/05	Tokyo Olympics proceed without spectators.
2021/06	Delta variant spreads rapidly worldwide.
2021/07	The highly transmissible Delta variant caused a rapid increase in COVID-19 cases worldwide. Hospitals in many countries, including the U.S., India, and Indonesia, were overwhelmed, leading to rising fears about the variant's impact on vaccine effectiveness.
2021/08	Reports highlighted the stark inequality in vaccine distribution, with wealthy countries administering booster shots while poorer nations struggled to vaccinate even frontline workers. This fueled global criticism and fear of prolonged pandemic impacts.
2021/09	The World Health Organization (WHO) classified the Mu variant (B.1.621) as a "variant of interest." Concerns grew about its potential to evade immunity from prior infections or vaccinations, adding to global anxiety.
2021/10	WHO warns of slow vaccination rates in Africa.
2021/11	Omicron variant identified in South Africa.
2021/12	Omicron-driven surge overwhelms global healthcare systems.
2022/01	COVID-19 cases reach record highs globally.
2022/02	Russia invades Ukraine, complicating pandemic recovery efforts.
2022/03	Shanghai enters strict lockdown amid China's zero-COVID policy.

Table 20: Key events related to the COVID-19 pandemic, covering major global developments from the early stages of the outbreak through the challenges of new variants and the ongoing efforts for pandemic recovery.

Time	News
2/13	Russian forces launched a full-scale invasion of Ukraine, marking the beginning of the most intense phase of the conflict. The attack included airstrikes, ground invasions, and naval assaults targeting major Ukrainian cities, including Kyiv, Kharkiv, and Odessa.
2/14	Ukrainian President Volodymyr Zelenskyy rejected an offer of evacuation from the U.S., stating that he needed ammunition, not a ride. Ukrainian forces fiercely resisted Russian advances despite being outnumbered.
2/15	Western countries, including the U.S., European Union, and the UK, imposed heavy sanctions on Russia, targeting banks, businesses, and prominent individuals. NATO countries began sending weapons and supplies to Ukraine.
2/16	Russian forces took control of the Chernobyl nuclear power plant, which had been the site of a catastrophic nuclear disaster in 1986. This raised fears of a nuclear incident amid the ongoing conflict.
2/17	The UN held an emergency session in response to Russia's invasion, with many countries condemning the aggression. Russia vetoed a resolution that would have demanded a ceasefire and withdrawal of forces from Ukraine.
2/18	Russian troops moved closer to Kyiv, Ukraine's capital, while intensifying their assault on cities in eastern Ukraine. Meanwhile, Russia announced it was placing its nuclear forces on alert.
2/19	Ukraine formally applied for European Union membership, emphasizing its desire to align more closely with Western Europe and away from Russian influence.
2/20	Thousands of Ukrainians fled westward to neighboring countries, especially Poland, as the war caused a massive refugee crisis. Humanitarian aid efforts ramped up, though conditions remained dire in besieged cities.
2/21	Russian forces continued to move toward Kyiv, and the city became a focal point of fierce fighting. Ukrainian President Zelenskyy remained in Kyiv, despite calls for his evacuation.
2/22	Ukrainian cities, including Mariupol, faced severe bombardment. Reports began emerging of significant civilian casualties and destruction due to Russian artillery and airstrikes.
2/23	Russian troops effectively encircled Mariupol, a port city in southern Ukraine, cutting off supplies and trapping thousands of civilians.
2/24	NATO leaders met to discuss increased defense aid for Ukraine, while the EU announced new sanctions against Russia, including restrictions on its access to financial systems and technology.
2/25	The international community, including the UN, continued to condemn Russia's actions. Reports of Russian war crimes, including targeting civilians and hospitals, emerged from various parts of Ukraine.
2/26	Humanitarian aid convoys attempted to reach the city, but Russian forces blocked routes, continuing their siege. Meanwhile, the UN confirmed over 2 million refugees had fled Ukraine.
2/27	Despite heavy bombardment, Ukrainian forces continued to put up strong resistance in Kyiv, Kharkiv, and other cities, using guerrilla tactics and fighting house to house.
2/28	The UN General Assembly passed a resolution demanding Russia cease its invasion of Ukraine, with a significant majority of countries voting in favor, though Russia and a few allies opposed it.
3/1	Russian troops captured large parts of southern Ukraine, including the city of Kherson, which became the first major city to fall under Russian control.
3/2	Russia continued its military advance, focusing on strategic locations like Mariupol, which remained besieged, while fighting continued on multiple fronts, especially in the Donbas region.

Table 21: Timeline of key events during the early stages of Russian-Ukraine Conflict in 2022.

Comparison	COVID-19 Δ_{Bias}	COVID-19 Δ_{Div}	RU-UA Δ_{Bias}	RU-UA Δ_{Div}
EvoBot vs. BC	< 0.05 (*)	< 0.001 (***)	0.412 (ns)	< 0.001 (***)
EvoBot vs. Lorenz	< 0.01 (**)	< 0.001 (***)	< 0.001 (***)	< 0.001 (***)
EvoBot vs. Llama2	< 0.01 (**)	< 0.001 (***)	< 0.001 (***)	< 0.01 (**)
EvoBot vs. GPT-4o	0.185 (ns)	< 0.05 (*)	< 0.05 (*)	< 0.05 (*)

Table 22: Statistical significance of EvoBot's simulation performance against baselines using the Mann-Whitney U test. Significance levels are: ns $p \geq 0.05$; * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$.

Simulation	Metric	Llama2-7b	GPT-4o-mini	EvoBot (Ours)
COVID-19	Δ_{Bias}	0.096 ± 0.005	0.083 ± 0.004	$\mathbf{0.074 \pm 0.003}$
	Δ_{Div}	0.107 ± 0.006	0.085 ± 0.005	$\mathbf{0.054 \pm 0.004}$
RU-UA Conflict	Δ_{Bias}	0.205 ± 0.010	0.132 ± 0.008	$\mathbf{0.103 \pm 0.006}$
	Δ_{Div}	0.261 ± 0.012	0.240 ± 0.011	$\mathbf{0.197 \pm 0.009}$

Table 23: Stability of key simulation metrics across three runs, showing mean and standard deviation.