Behind the Chat: Agent-Based Modeling of Twitch Mod Bots vs. Spambots

GitHub Repository URL: https://github.com/Peterayade/EECS-4461-Project.git

Team Information

- **Team Number**: Team 5
- **Team Members**:
  - Peter AYADE
  - Mingran CHEN
  - Shenice THOMAS

§1. Phenomenon Overview

The chosen phenomenon involves the escalating interaction between spambots and moderation bots (mod bots) on live-streaming platforms such as Twitch. This AI-to-AI interaction represents a technologically evolving arms race, where adversarial (spambots) and regulatory (mod bots) agents continuously adapt their behavior to outmaneuver one another in real time (Calvaresi et al., 2023). Spambots bypass detection by mimicking human behavior, modifying message timing, tone, and structure, while mod bots respond with increasingly advanced moderation techniques to maintain order in the chatroom. This interaction is highly significant within media ecosystems because it directly impacts user trust, platform engagement, and the integrity of Twitch and even other platforms' online discourse. As moderation systems become more automated, they introduce new capabilities but also expose vulnerabilities. Failures in moderation can lead to an unchecked spread of spam and misinformation or, conversely, result in the unfair censorship of legitimate users due to algorithmic bias (Wohn et al., 2023). These dynamics raise broader concerns around AI governance, transparency, and fairness in digital environments (Cai & Wohn, 2019).

The core issue lies in the imbalance and unpredictability caused by rapidly evolving adversarial AI. As spambots become smarter, traditional moderation becomes less effective, often resulting in either false negatives (spambots evading bans) or false positives (real users being banned). This weakens platform governance and user trust. The situation becomes more complex when audience members, influenced by high spam exposure, begin amplifying spam themselves, destabilizing the system further (Calderini et al., 2022). This is not just a Twitch-specific issue; platforms like YouTube, Reddit, and X face similar dynamics, where manual moderation is infeasible and automated systems must tread a fine line between effectiveness and overreach. These interactions create a self-reinforcing feedback loop, where decisions by one AI type influence the evolution of the other, leading to emergent and often unintended consequences.

Agent-Based Modelling (ABM) is uniquely suited to studying this phenomenon. It allows us to simulate complex, adaptive, and autonomous agents, spambots, mod bots, and audience members, each operating by localized rules in a shared environment. ABM supports dynamic interaction simulation, capturing the evolution of behavior over time; emergent behavior analysis, revealing unexpected system-wide patterns; and testing of adaptive strategies, enabling experimentation with moderation rules, bot IQ levels, and spam thresholds. Its scalability and realism allow micro-level interactions to reflect macro-level outcomes (Calvaresi et al., 2023; Cai & Wohn, 2019), making it more effective than rigid, rule-based models in capturing the nonlinear, feedback-driven nature of

AI moderation in media ecosystems.

§2. Simulation Design & Implementation

System Overview:

We modified the Epstein Violence Model in the mesa model to simulate the behavior and interaction of modbots, spambots, and audiences in the Twitch live broadcast room. The phenomena we are interested in include how spam bots determine the timing of spam activities, the confrontation between modbots with different intelligence levels and spam bots with different intelligence levels, and the audience's reaction to the information in the live broadcast room.

After testing and analysis, we found that after the model runs for a period of time, the number and proportion of agents in each state of the live broadcast room will tend to a stable state, but there will still be sudden and drastic changes occasionally (for example, the number of agent accounts of acitve suddenly increases significantly).

Simulation Environment:

The model we adopted is a grid-based model. In the environment we set, each agent occupies a grid. Each agent will act and react to the environment.

The following are important controllable environmental parameters of the model:

[1. Initial Agent Density]

The initial agent density. This is used to control the number of accounts in the current live broadcast room, that is, the number of agents in the model.

[2. Spambot ratio]

Used to control the ratio of spambot accounts, to simulate the impact of different numbers of spambots on the model. The minimum value of this parameter is 10%, which means that at least 10% of the accounts are spambots; the maximum value is 90, which means that at most 90% of the accounts are spambots.

[3. Initial Modbot Density]

This parameter represents the proportion of modbot accounts. It is used to simulate the impact of different numbers of modbots on the live broadcast room and model. The proportion range is 0%-10%.

[4. Spambot Vision]

This parameter represents the perception ability of the spambot. The larger the parameter, the stronger the processing ability of spambot. Spambot can judge whether to conduct spam activities based on the status of more accounts. The parameter range is 1 grid-10 grids.

[5. Modbot Vision]

This parameter represents the perception ability of modbot. The larger the parameter, the stronger the processing ability of modbot. Spambot can judge whether to conduct spam activities based on the status of more accounts. The parameter range is 1 grid-10 grids.

[6. Registration difficulty]

This parameter represents the difficulty of registering platform accounts. In addition to banning spam_bot, network platforms will also take a series of measures to increase the difficulty of creating spam_bot accounts, such as requiring accounts to bind mobile phone numbers or email addresses, using verification codes, etc.

Therefore, the higher the registration difficulty, the higher the cost of spam_account, and they will be more cautious in making decisions to prevent being banned by modbot.

[7. Max ban time]

This parameter represents the maximum time modbot bans an account (except for permanent bans).

[8. Spambot IQ Range]

This parameter represents the upper limit of spambot's intelligence level. The larger the parameter, the higher the possible intelligence level of spambot, and the more likely it is to escape modbot's detection and ban.

[9. Modbot IQ]

This parameter represents the intelligence level of modbot. The higher the parameter, the higher the intelligence level of modbot, and the easier it is to detect spambot and ban it.

[10. Times of ban before forever ban]

This parameter represents the number of times an account is banned forever. When the number of bans equals this parameter, the account will be permanently banned and will not be unbanned again. When this parameter is set to -1, it means that modbot will never ban an account forever.

Agent Design:

In our model, there are three agents in total:

[1. Spambot]

Spambot will calculate its spam tendency based on the difficulty of account registration, the spam demand of the spambot account (a built-in parameter of the Spambot Agent, which represents the operator's demand for spambot activities) and the risk judgment of spambot. When the spam tendency is higher than its threshold, the spambot will decide that it is suitable for activities now.

The key issue when instantiating Spambot is how to reasonably design its decision-making system and how to set various parameters.

[2. Modbot]

Modbot will patrol the live broadcast room, identify the identity of the account in the field of view, and ban the active account that is currently engaging in spam activities. When Modbot makes a decision on whether to ban, it will consider its own intelligence level and the intelligence level of the suspicious account. When Modbot's intelligence level is higher than the intelligence level of the suspicious account, it will make the right decision, otherwise it will make the wrong decision.

The key issue when instantiating Modbot is how to reasonably design its intelligence level and decision-making system.

[3. Audience]

Audience itself will not take any initiative to perform any actions, but will only remain in a quiet state to simulate ordinary audiences. However, when the number of active spaming accounts around an Audience is higher than a certain level, the audience will be confused and inadvertently turn into an active state. The purpose of this is to simulate the situation where ordinary people are deceived and forwarded by disinformation or missing information. When instantiating Audience, the key issue is how to design the mechanism for the audience to be deceived.

Interaction Dynamics:

The mesa model adopted by this model is Epstein Violence scheduling. The reason why we adopt this scheduling strategy is that its arrest mechanism and riot propagation mechanism are very suitable for simulating the confrontation between Modbot and Spambot in Twitch live broadcast rooms and the impact of Spambot on ordinary audiences.

The bot-to-bot interactions between Spambot and Modbot are in two aspects:

1. Spambot will judge whether it is the right time to conduct spam activities based on the number and risk of Modbots around itself. When Spambot finds that there are too many modbots around it, it will think that the risk is too high and it is not suitable to carry out activities, so it will remain quiet.

2. Modbot interacts with Spambot in two ways. First, modbot will try to identify the identity of spambot and try to block the spambot that is in an active state. The result of this interaction depends on the intelligence level of both parties. Second, if Modbot is deceived by Spambot in the interaction, it will learn from the failure and improve its intelligence level.

Human-to-bot interaction is the interaction between audience, spambot and modbot.

Audience will take actions due to the interaction between the information around it and its interaction. In other words, spambot may deceive Audience to participate in spam activities. And modbot will also block these audiences who accidentally participate in spam activities.

Therefore, the model's simulation of the interaction dynamic of bot-to-bot and human-to-bot well restores the interaction between different agents in the real-world twich live broadcast room.

Data Collection and Visualization:

Our model can observe and collect the following data through visualization:

1. The current number of different agents.

2. The specific number of agents in different states (audience quiet, spambot quiet, active, banned, banned forever).

3. The proportion of agents in different states.

4. The change of agent states over time.

Colour explanation: In our visualization component, blue represents spambot in quiet state, green represents audience in quiet state, black represents modbot, gray represents temporarily banned accounts (including spambot and audience), and purple represents permanently banned accounts.

Grid graph:

Grid graph can clearly show us the status and number of each agent in the current simulation system.

Plot graph

The purpose of the plot graph is to depict the changing trends of agents in different states, making it easier for us to observe the mutual influence of interactions between agents.

Pie chart:

The purpose of the pie chart is to provide real-time feedback on the proportion of agents currently in different states.

The key issue encountered in the visualization process is that a large number of different agents and their status need to be aggregated and analyzed, and then assigned to different visualization modules. This process is quite difficult, and program errors often occur, resulting in inconsistent data between visualization modules. This key issue has been resolved.

§3. Observations & Results

Simulation results:

The main purpose of this model is to simulate the interaction of Spambot, Modbot, and Audience in Twitch live broadcast rooms and the impact of different environmental variables on the interaction. We control different variables and compare and test the different effects of different factors on the live broadcast room. The results we mainly focus on are as follows:

1. The time required for the model to stabilize.
2. The frequency of sudden large changes after the model is stable.
3. The magnitude of sudden large changes after the model is stable.
4. The proportion of different states when the proportion of different agents in the live .broadcast room tends to stabilize.

All parameters will have different effects on the simulation of the model to a greater or lesser extent. After many tests, among all the parameters, the registration difficulty, Spambot ratio and Times of ban before forever ban have a much greater impact than other parameters. The following will focus on the impact of these parameters and the phenomena they simulate:

Only change one variable at a time, run 200 steps. The default values of each setting variable are as follows:

Initial Agent Density: 0.7

Spambot ratio: 70

Initial Modbot Density: 0.04

Spambot Vision: 7

Modbot Vision: 7

Registration difficulty: 0.8

Max ban time: 30

Spambot IQ range: 0.7

Modbot IQ: 0.7

Times of ban before forever ban: 7

1. Change registration difficulty

High registration difficulty

Low registration difficulty

Based on the observed results, we found that the greater the registration difficulty, the shorter the time it takes for the system to stabilize, and the lower the frequency and magnitude of subsequent emergencies. The opposite is true. The greater the registration difficulty, the more spambots will value their accounts and the more cautious they will be. This also results in fewer accounts being permanently banned.

2. Change the Spambot ratio

High spambot ratio

Low spambot ratio

Based on observations, we can analyze that when the proportion of spambots in the live broadcast room system is small, the system will stabilize faster, and the frequency and magnitude of emergencies will be smaller. This is because when the number of spambots is small, the audience is not easily deceived by spambots.

3. Change Times of ban before forever ban

An account will be permanently banned only after it has been banned lots of times.

When an account is banned twice, it will be banned immediately and permanently.

After analysis, this parameter has a huge impact on all results. When this parameter is small (1-5), Modbot will ban a large number of accounts in a short period of time. This makes the time for the live broadcast room to reach a stable state very short. At the same time, after the live broadcast room reaches stability, the interval between fluctuations is longer and the fluctuation amplitude is smaller.

When this parameter is large, Modbot takes longer to ban accounts. This means that the system takes longer to reach a stable state. At the same time, this also means that the frequency of fluctuations is fast, and the amplitude is higher.

Unexpected behaviors

As we mentioned in the presentation and Deliverable 3, we did not expect the system to still fluctuate after stabilizing. Our original guess about the system was that after modbot banned spambots and the accounts of audiences who accidentally participated in spam activities, the agents in each state would maintain a constant ratio that would never change.

However, the actual situation we observed was that after the system was in a stable state for a period of time, a large number of agent accounts suddenly turned into active states, which means that a large number of accounts suddenly engaged in spam activities. And the interval time and change amplitude of such sudden situations are regular.

As shown in the figure below, after 25 steps, the model has tended to a relatively stable state, but the plot still fluctuates. In particular, there are large changes at about 30, 60, 100, 125, 175, and 200 steps.

After analysis, the possible reasons for this phenomenon are as follows:

1. Modbot banned many spambot accounts at the same time, and their ban time was exactly the same. This means that many spambot accounts will be unbanned at the same time.
2. Many spambots turned to an active state at the same time, which means that a large amount of spam information will suddenly appear in the live broadcast room. This will cause many audiences to mistakenly believe this information and participate in spam activities. This, in turn, will lead to an increase in spam information. Such a domino effect has led to emergencies.

§4. Ethical & Societal Reflections

Ethical Considerations

Our project simulates the interaction between modbots and spambots in a Twitch-like media environment. While the simulation does not directly incorporate real-world data sets such as chat logs, it shows behavioural patterns from real live-streaming platforms. Although our project did not raise privacy concerns through data collection, it touches several important ethical themes around data usage, surveillance and algorithmic fairness.

To begin with, our model assumes that modbots are continuously monitoring all chat behaviour in real time. This raises privacy concerns, particularly when considering that users may not be fully aware of how much their communication is being analyzed, flagged or recorded by moderation algorithms. It can be hard to tell where moderation ends and surveillance begins, especially when the system looks at users' behaviour, message tone, or how often someone chats.

Moreover, our model assumes that bots can detect "spam-like" patterns, which in real-world scenarios often include keyword matching or machine learning classifiers trained on historical chat data. This presents a risk of algorithmic bias, especially toward users who use non-standard language, slang or speech patterns common in marginalized communities. Without inclusive training data, modbots may disproportionally flag innocent messages, leading to false bans and causing users to feel unsure or uncomfortable using the platform.

To avoid unfair treatment, platforms should use clear moderation policies, include human viewers for appeals, and train AI using diverse language styles, cultures, and social behaviours.

Societal Implications

At a micro-level, our findings highlight how individual users can be unfairly impacted by automated moderation systems. Quiet spambots can mimic real users and avoid detection, while legitimate users might be flagged based on superficial message features. This reflects real-world moderation failures, such as Twitch's occasional bans of innocent users during automated enforcement sweeps.

At a meso-level, we can see an emergent arms race between AI systems; modbots adapt to ban more effectively, while spambots evolve to evade them. This reflects real-world developments where bot creators continuously refine evasion tactics and platforms escalate detection strategies using natural language processing, pattern analysis, and machine learning. The result is a feedback loop where both sides escalate, but the platforms may become increasingly reliant on opaque, complex algorithms to maintain order, potentially reducing transparency and user agency.

At the macro-level, the constant back-and-forth between spambots and modbots shows how power is changing online. While human moderators can make decisions with care and understanding, sometimes these choices are being made by automated systems that work fast but lack empathy or cultural awareness. These systems can influence what people see and say online, and sometimes they accidentally silence voices that don't fit the "norm," even if that wasn't the goal.

Our simulation aligns with real-world moderation challenges in that modbots tend to be more reactive than proactive, often lagging behind new spambot strategies. Just as in our simulation, real-world moderation rarely achieves total spam elimination and instead settles into a fluctuating equilibrium tolerating some level of spam in exchange for scalable management.

Though our simulation is designed to explore moderation strategies, its framework could be repurposed unethically, for example:

- A malicious actor could adapt the spambot agents to study how to evade detection across different environments.
- A platform operator could misuse how modbots work to unfairly silence certain people or opinions, especially if they program the bot to look for behaviours or language styles that only some groups use.
- Governments or authoritarian regimes could apply similar situations to optimize social media censorship, training algorithms to silence speech under the guise of "harmful content".

Therefore, these concerns emphasize the importance of AI governance and ethical safeguards, particularly around how simulation tools are shared, deployed, and extended.

To conclude, our simulation contributes to the understanding of automated moderation dynamics but also surfaces deeper concerns around surveillance, bias, and speech governance in AI-driven platforms like Twitch. While simulations like ours can guide better design, they should not only serve efficiently but also be accompanied by ethical frameworks, transparent policies, and human oversight to ensure they serve the public good.

§5. Lessons Learned & Future Directions

Design and Development Reflections

One of our primary challenges was modifying the Epstein Violence Model to simulate a Twitch-like live chat environment. Originally designed to model civil violence, the Epstein model did not account for the nuanced interactions among different types of agents, spambots, modbots, and audience members. Our team had to redefine agent behaviours: spambots now calculate a "spam tendency" based on parameters such as spam demand and registration difficulty. At the same time, mod bots engage in an AI-to-AI arms race using IQ comparisons and learning mechanisms. For example, when a mod bot misidentifies a spambot, its intelligence is incrementally improved, reflecting an adaptive learning process miming real-world AI evolution.

A key aspect of our agent design is the explicit decision-making logic implemented for spambots. The logic is defined as follows:

$$\text{spam\_tendency} = \text{spam\_demand} \times (1 - \text{registration\_difficulty})$$

Spambots actively post spam only if their calculated spam_tendency exceeds a set threshold. Otherwise, they remain quiet to avoid bans. Further, the decision rule adjusts for risk by considering the formula:

$$\text{if } (\text{spam\_tendency} - (\text{risk\_aversion} - \text{ban\_probability}) > \text{threshold})$$

This succinct representation clarifies how each spambot decides when to act and directly ties the simulation's parameters to emergent behaviours in the model.

Implementing these dynamics using the Mesa framework presented additional challenges. Mesa's Epstein violence scheduling, with its arrest and riot propagation mechanics, was ideal for simulating bot confrontations. However, the scheduling system's complexity demanded careful calibration. Early on, our approach to modelling agent intelligence using separate high-IQ and low-IQ classes proved computationally expensive and inflexible. We resolved this by streamlining intelligence into a single behavioural attribute, maintaining meaningful distinctions between agents while reducing overhead. Furthermore, integrating real-time learning for mod bots triggered upon incorrect bans required extensive testing and parameter tuning to ensure system stability.

Visualization was another hurdle. Our initial choice of SolarViz was fraught with compatibility issues, resulting in delays and inconsistent data across visualization modules. Switching to Mesa's built-in tools allowed us to deploy grid graphs, plot graphs, and pie charts, although this came at the cost of some visual richness. We ultimately resolved data synchronization issues through iterative debugging, ensuring that each visualization accurately reflected the agents' states and interactions.

Model Limitations & Areas for Improvement

Despite our successes, several oversimplifications constrain the simulation's fidelity to real-world phenomena. One notable limitation is the static treatment of agent intelligence. While our mod bots can learn from their mistakes, resulting in fewer false negatives when detecting smarter spambots, the intelligence levels of spambots remain fixed. This static approach leads to occasional false positives where legitimate users are mistakenly banned and false negatives, where more sophisticated spambots evade detection. Furthermore, the model's assumption of uniform ban durations and simplified unbanning cycles has resulted in phenomena like synchronized reactivation of banned accounts, an artifact that, while insightful, does not fully mirror the unpredictable nature of human behaviour in live chats.

Another area for improvement is the binary classification of audience behaviour. Currently, audience members are modelled as passive or fully engaging in spam when exposed to a high spam-to-normal message ratio. This oversimplification ignores the broad spectrum of human reactions, limiting the model's ability to simulate realistic behaviour under varying conditions.

Future Refinements and Applications

Several enhancements could address these limitations. A key refinement would involve developing dynamic learning algorithms for mod bots and spambots, allowing their intelligence levels to evolve continuously in response to adversarial interactions. Implementing a more nuanced "ban_forever" mechanism triggered after a calculated number of offences could better mirror real-world moderation policies. Additionally, integrating real-world datasets such as historical chat logs could help refine parameter settings and improve the model's fidelity. On the visualization front, future work might include integrating advanced tools like interactive dashboards or network graphs to capture and analyze complex agent interactions more effectively. These enhancements would provide deeper insights into emergent patterns and facilitate more responsive system adjustments. The insights from our simulation extend well beyond academic inquiry. They have practical applications in platform governance, informing policies that balance effective moderation with user engagement. Our findings could help develop safer, more adaptive moderation systems that minimize false positives and negatives. In the broader context of AI safety research, our project underscores the importance of incorporating adaptive learning and realistic behavioural dynamics into simulation models, ultimately contributing to the design of fairer and more resilient digital platforms.

§6. References

Calvaresi, D., Dubovitskaya, A., Taveter, K., Schumacher, M., & Främling, K. (2023). Exploring agent-based chatbots: A systematic literature review. Journal of Ambient Intelligence and Humanized Computing, 14(8), 11207–11226. https://doi.org/10.1007/s12652-023-04626-5

Cai, J., & Wohn, D. Y. (2019). Categorizing live streaming moderation tools: An analysis of Twitch. International Journal of Interactive Communication Systems and Technologies, 9(2), 36–50. https://doi.org/10.4018/IJICST.2019070103

Caldarini, G., Jaf, S., & McGinnity, T. M. (2022). A literature survey of recent advances in chatbots. Information, 13(1), 41. https://doi.org/10.3390/info13010041

Steed, R., & Caliskan, A. (2021). Image representations learned with unsupervised learning contain human-like biases. In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT '21) (pp. 701–713). Association for Computing Machinery. https://doi.org/10.1145/3442188.3445922

Whittaker, M., Crawford, K., Dobbe, R., Fried, G., Kaziunas, E., Mathur, V., West, S. M., Richardson, R., Schultz, J., & Schwartz, O. (2018). AI Now Report 2018. AI Now Institute. https://ainowinstitute.org/AI_Now_2018_Report.pdf

§7. Attestation

Mingran Chen:

- **Writing – Original Draft**: Took lead on the versions of core simulation content, including Simulation Design & Implementation (Part 2), Observations & Results (Part 3) as well as drafting Lessons Learned & Future Directions (Part 5) for previous deliverables.
- **Writing – Review & Editing**: Acted an editor for the final report, merging contributions from all team members, standardizing tone and formatting, and ensuring narrative consistency throughout.
- **Visualization**: Managed integration of visual assets (graphs, simulation screenshots, and data plots), and helped interpret their relevance to the findings.

Peter Ayade:

- **Conceptualization**: Contributed to the overall research focus, aligning the project goals with a Twitch-like environment and guiding the adaptation of the Epstein Civil Violence Model.
- **Writing –Final Draft**: Authored initial versions of key sections, including the introductory overview (Part 1) and Lessons Learned & Future Directions (Part 5).
- **Writing – Review & Editing**: Served as the primary editor, formatting the entire document for clarity and consistency, merging contributions from all team members, and finalizing the cohesive narrative of the final report.

Shenice Thomas:

- **Writing - Original Draft**: Contributed to the Ethical and Societial Reflections (Part 4) and provided stutured analysis with real-world connections and citations.

- **Data Curation & Documentation**: Created and formatted the Markdown (.md) version of the final report, ensuring structure and consistency across sections.
- **Software**: Set up and maintained the GitHub-based automated PDF generation, testing, and resolving layout issues to ensure the final report PDF was correctly formatted.