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

Phenomenon Overview

Live-streaming platforms like Twitch rely on automated moderation systems to manage chat interactions, but spambots continuously evolve to evade detection, forcing mod bots to adapt in real-time (Wohn et al., 2023). As live broadcasting grows in popularity, fraudulent activities have followed, with spambots spreading false or hateful information by exploiting platform-specific parameters. To counter this, Twitch has implemented mod bots to regulate chatrooms, but this remains an ongoing challenge, as spambots refine their evasion techniques while mod bots adjust their detection algorithms (Calvaresi et al., 2023). This AI-to-AI interaction is critical to media ecosystems, where moderation must keep pace with adversarial AI. Spambots not only degrade user experience but also manipulate engagement metrics and spread misinformation, while mod bots must balance enforcement without mistakenly restricting real users. Beyond Twitch, these challenges extend to social media platforms and online forums, making AI moderation a crucial area for further research (Cai & Wohn, 2019).

The arms race between mod bots and spambots presents a significant challenge, as spambots modify messages, mimic human behaviour, and adjust spam frequency to evade detection. When moderation fails, false positives occur when legitimate users are mistakenly flagged and banned, leading to frustration and reduced engagement, while false negatives allow spambots to remain undetected, flooding chat rooms with unwanted content and manipulating conversations. Both outcomes undermine platform integrity and user experience, emphasizing the need for a more adaptive and precise moderation system.

Agent-Based Modeling (ABM) provides an ideal framework for understanding and improving AI-driven moderation by allowing researchers to simulate how mod bots and spambots interact over time. Dynamic Interaction Simulation enables ABM to model these entities as autonomous agents that evolve in response to each other, mimicking real-world AI adaptation. Emergent Behavior Analysis helps capture complex patterns in chat moderation, such as spambots modifying tactics to evade detection and mod bots refining filtering mechanisms accordingly. Testing Adaptive Strategies allows for experimentation with different moderation policies by adjusting mod bot strictness, spam detection thresholds, and learning mechanisms to measure their impact on chat regulation. Finally, Scalability and Realism make ABM a powerful tool by ensuring that individual agent decisions shape system-wide trends, reflecting real-world unpredictability better than rigid rule-based models. By leveraging ABM, we can better understand how moderation strategies influence chat quality, spam distribution, and user engagement, ultimately leading to more effective

and adaptable AI-driven moderation systems.

In the Twitch live broadcast platform, we can model the following roles as intelligent agents:

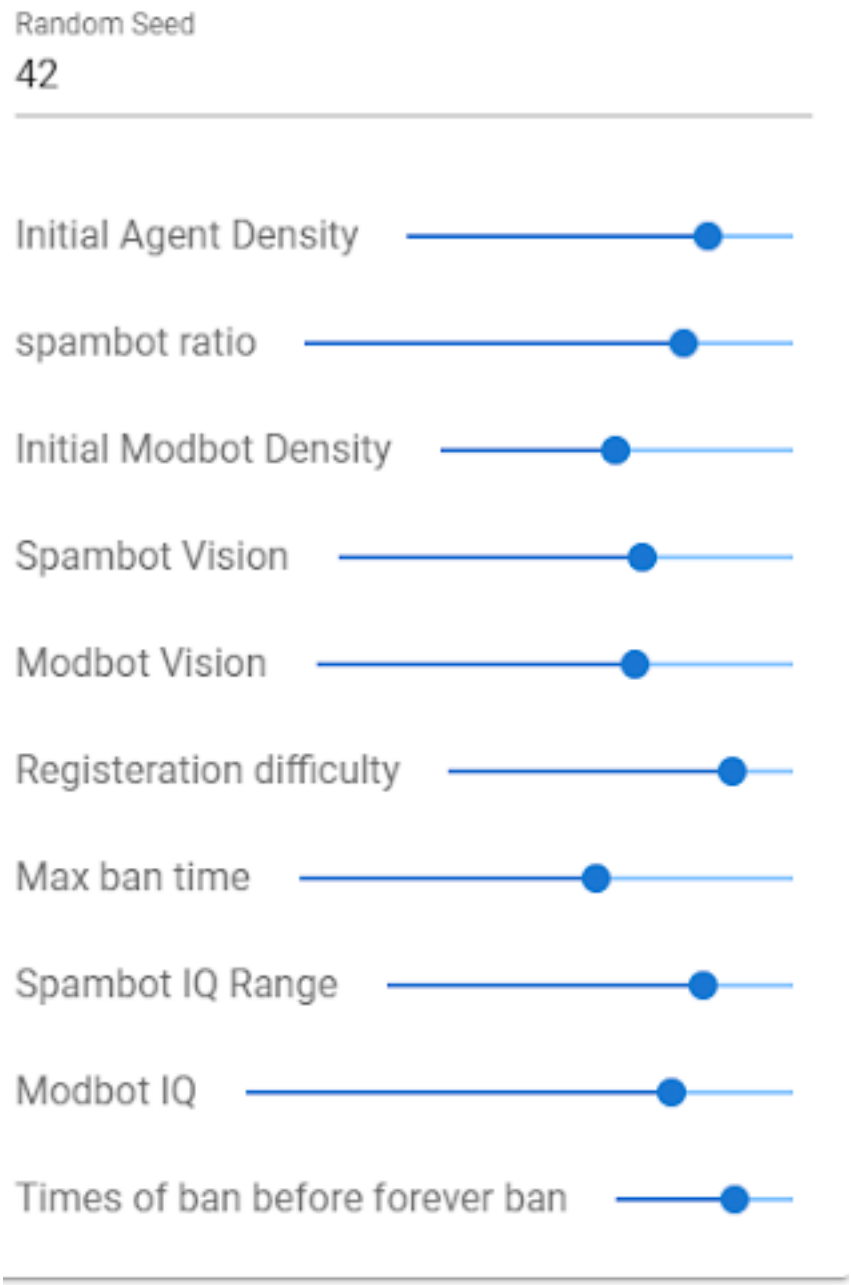
- **Spambot:** an automated AI that attempts to spread advertisements, hate speech or fraudulent information in the live broadcast room.
- **Modbot:** an AI regulator developed by Twitch, responsible for detecting and banning illegal Spambots.
- **Audience:** an ordinary user who may be disturbed by Spambot.

Visual Sequence

(Before Start)

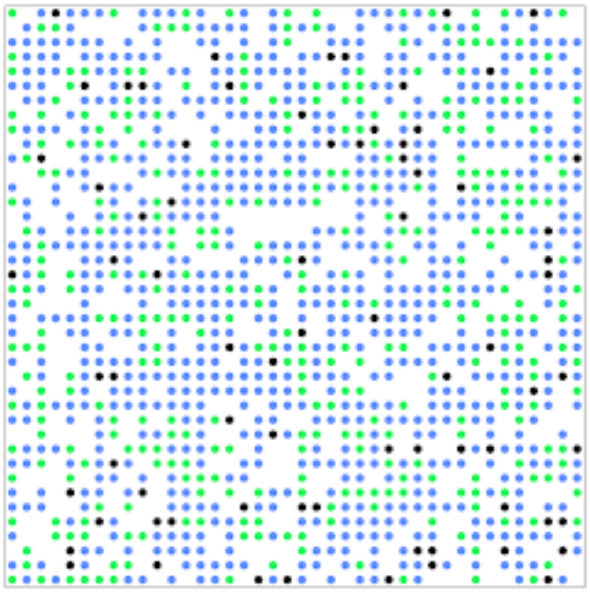
The parameter bar on the left is the various parameters set before running the simulation. Their detailed functions will be explained in the second part. This simulation contains three visual interfaces:

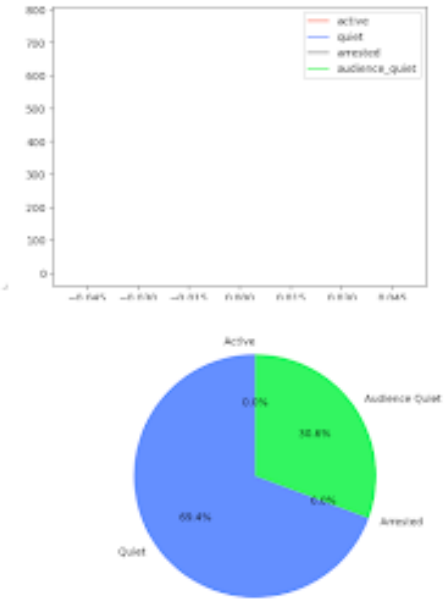
- **1. Agent grid:**
- **2. Agent plot chart:**
- **3. Agent pie chart**



The **agent grid** visualizes changes in agent behaviour based on the **modified Epstein Violence model**. The **agent plot chart** tracks agent activity over time, while the **pie chart** displays the real-time proportion of different agents. In the simulation, blue represents quiet spambots, **black** represents mod bots, **green** represents quiet audience members, **red** represents active spammers (spambots or audience), and **gray** indicates banned accounts.

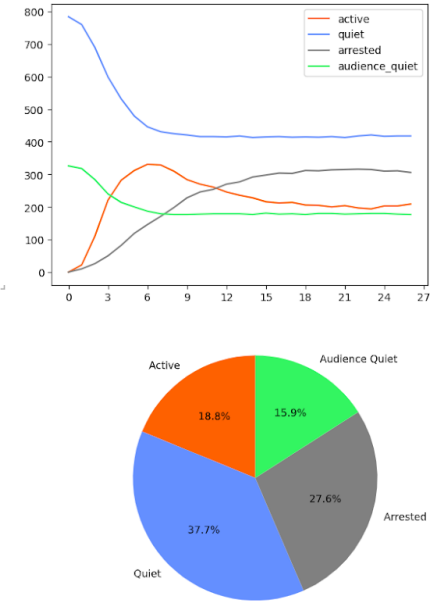
(Before running)



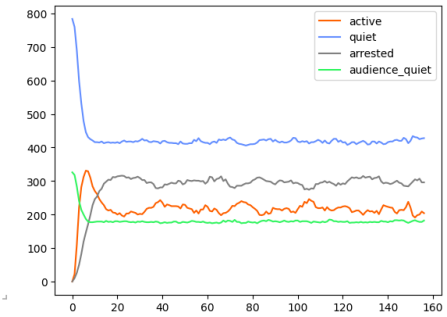


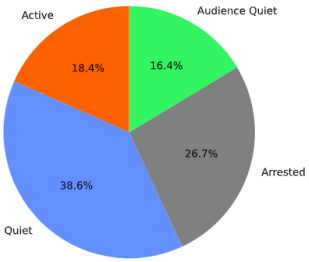
At the start, no accounts engage in spam, simulating the initial state of a newly opened streaming room.

(During running)



After the model starts running, it simulates the live broadcasting in the live broadcast room. Different agents take different actions to interact with each other, causing the situation in the live broadcast room to change.





We currently assume that the running time of the live broadcast room is usually 160 steps. We stop the run after 160 steps and observe the above results.

Simulation Design & Implementation

Our simulation models the interaction between spambots, mod bots, and audience members in a 40×40 grid-based live chat environment, mirroring real-world Twitch chatrooms. Spambots attempt to spread messages, mod bots enforce moderation, and audience members react dynamically. Various parameters Spam Demand, Risk Aversion, Legitimacy & Registration Difficulty, and Ban Probability shape agent behaviour, determining how spam spreads and is controlled over time.

System Overview & Agent Design

The model consists of three agent types, each following distinct behaviours and decision-making rules:

- Spambots determine whether to post spam based on:

spam_tendency=spam_demand×(1−registration_difficulty)

If spam tendency surpasses a threshold, they actively post spam; otherwise, they remain quiet to avoid bans.

If (spam_tendency - (risk_aversion - ban_probability)>therhold)

- Mod Bots patrol the chat, detecting spambots based on vision range and IQ comparison. If a mod bot's IQ is higher than the suspected spambot's IQ, it successfully bans the account; otherwise, the spambot evades detection. Bans may be temporary or permanent, depending on ban probability settings.
- Audience Members initially do not spam, but if exposed to a high spam-to-normal message ratio, they may unknowingly forward spam, simulating misinformation spread.

Simulation Environment & Interaction Dynamics

The 40×40 grid-based environment represents a live chatroom, where agents move and interact dynamically based on preset parameters and decision-making rules.

- **Scheduler**: The simulation uses StagedActivation, ensuring sequential decision-making:
  - Spambots decide whether to post messages based on spam tendencies.
  - Mod bots analyze the chat, detecting and banning spambots.
  - Audience members process the chat, reacting to spam levels.
- **Bot-to-Bot Interaction**:
  - Spambots adjust their spam patterns to evade bans.
  - Mod bots adapt their moderation techniques, improving spam detection over time.
  - Audience members may amplify or suppress spam, impacting chat dynamics.
- **Emergent Phenomena**: Over time, several key behaviours emerge:
  - Spambots modify message frequency and patterns to bypass bans.
  - Mod bots balance strictness vs. leniency, affecting chat engagement and spam suppression.
  - Audience behaviour shifts, either maintaining engagement or unintentionally amplifying spam.

Data Collection & Visualization

Our simulation collects real-time data on the number and proportion of spambots, mod bots, and audience members at each simulation step. Tracking agent status changes allows us to analyze how different parameters registration difficulty, spambot density, ban probability, and AI intelligence level affect spam spread and moderation effectiveness.

Visualizations include:

- Agent Grid – Displays spatial distribution and behaviour of agents in real-time.
- Line Charts – Track trends in spam activity, bans, and audience engagement over time.
- Pie Charts – Represent the proportion of active, banned, and quiet agents at different simulation steps.

Preliminary Observations & Results

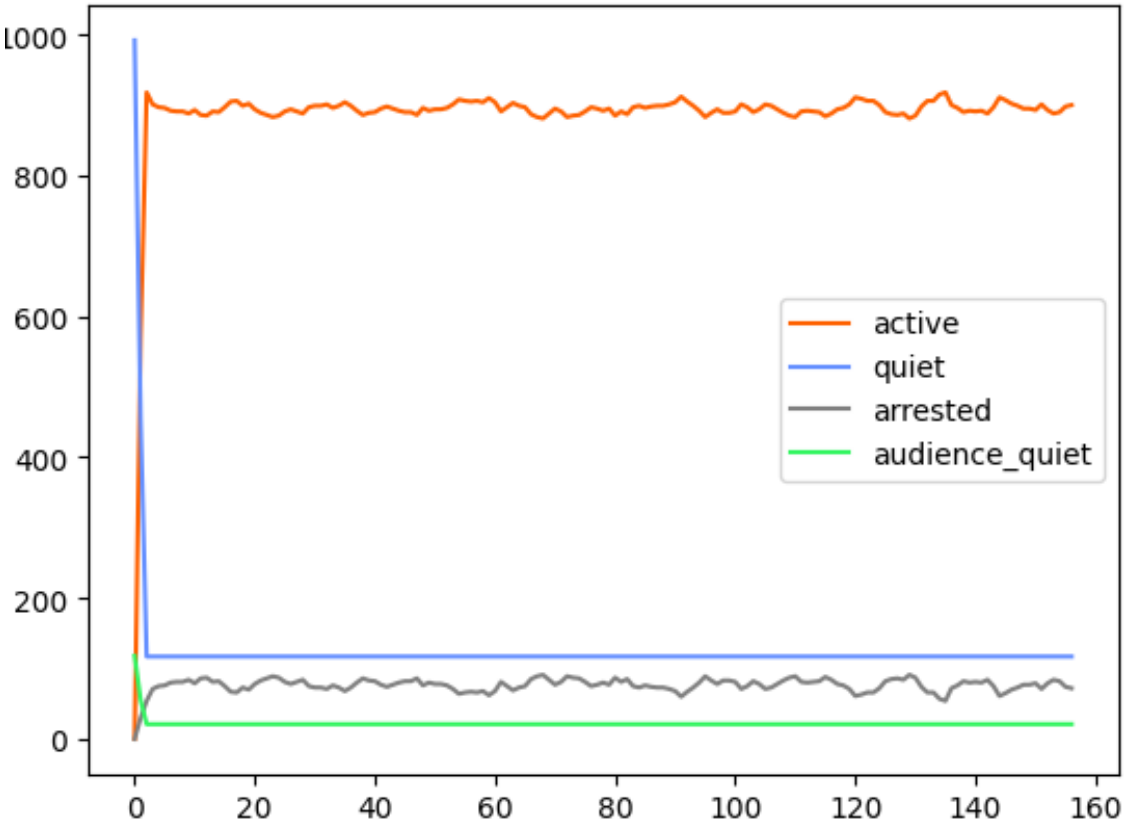
Over time, the number of agents stabilizes, creating a balance between mod bots, spambots, and audience members. However, stabilization speed and agent distribution vary depending on parameter settings. Key factors influencing stability include registration difficulty, max ban time, and spambot-to-mod bot IQ ratio. Higher registration difficulty reduces active spambots, longer ban times increase the number of banned accounts, and a higher spambot-to-mod bot IQ ratio results in more active spambots. Adjusting these parameters while keeping other settings constant leads to different model outcomes.

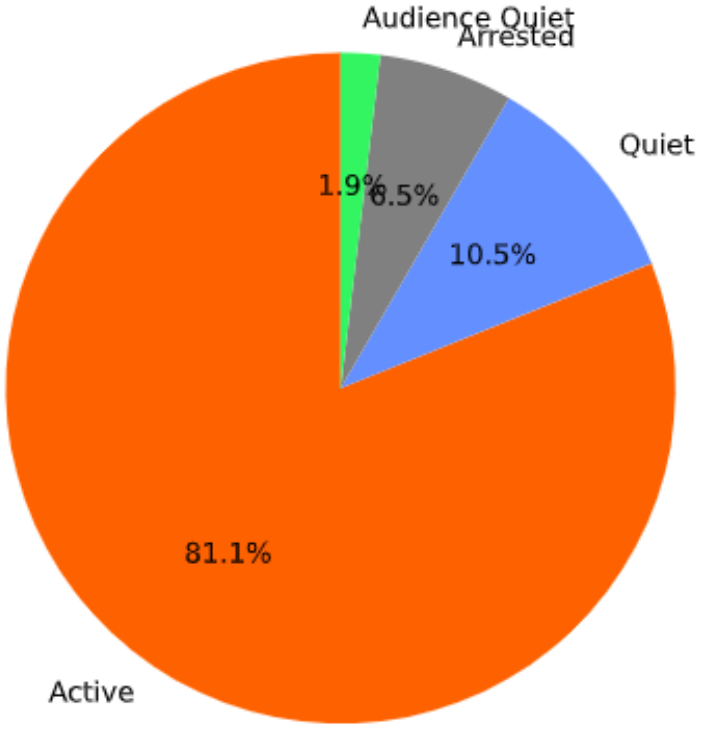
Comparison of Different Parameter Settings

Setting 1

- Low registration difficulty= 0.2

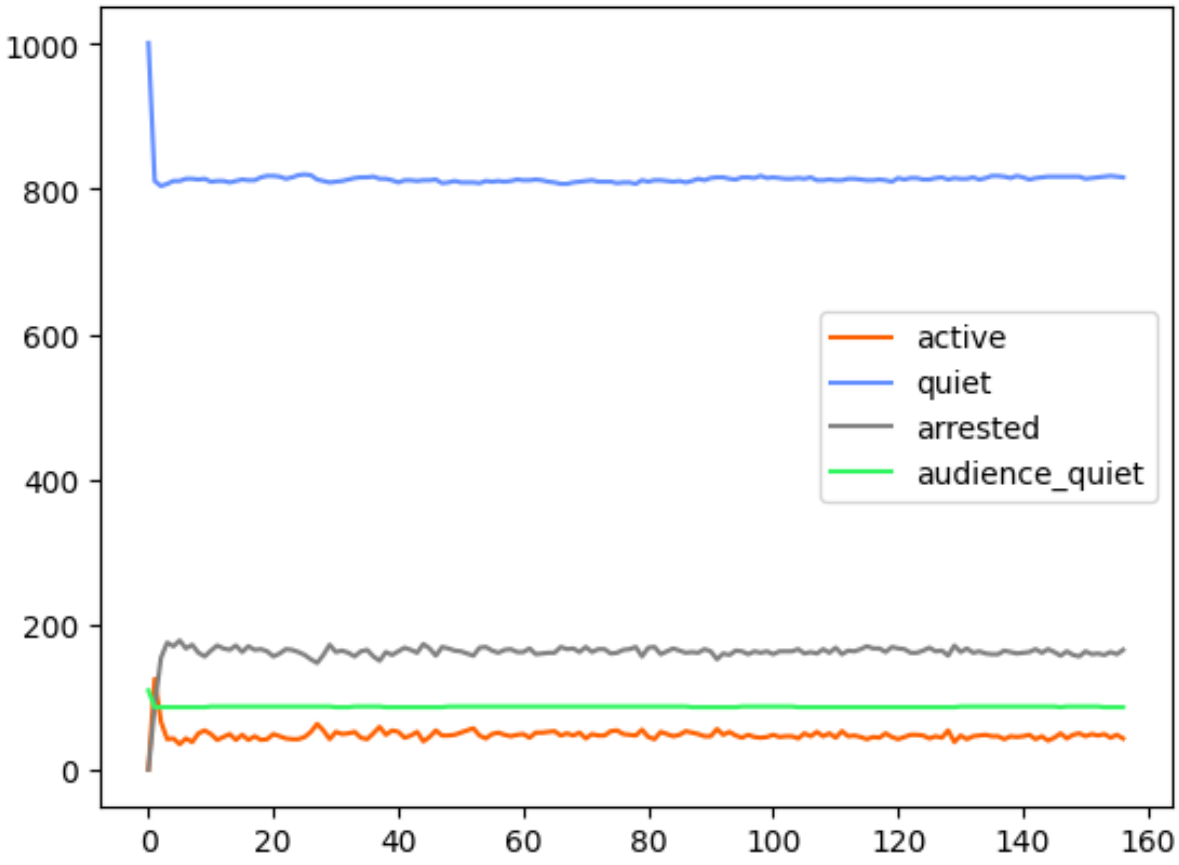
- Short max ban time =6
- Spambot IQ= 0.8 | Mod bot IQ= 0.2 -> Result: High spambot activity, low ban rates, ineffective moderation.

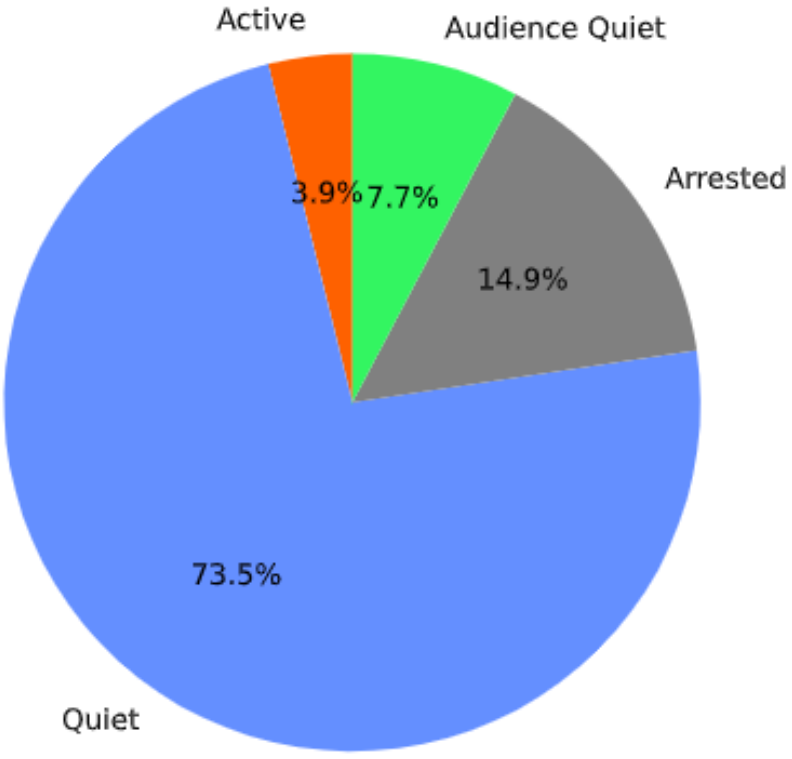

1000
800
600
400
200
0
0
20
40
60
80
100
120
140
160
active
quiet
arrested
audience_quiet


Audience Quiet
Arrested
Quiet
1.9%
8.5%
10.5%
81.1%
Active

Setting 2

- High registration difficulty: 0.8
- Long max ban time: 28
- Spambot IQ: 0.6
- Mod bot IQ: 0.8 -> Result: More bans, reduced active spambots, better moderation.


1000
800
600
400
200
0
0
20
40
60
80
100
120
140
160
active
quiet
arrested
audience_quiet


Active
Audience Quiet
Arrested
3.9%
7.7%
14.9%
73.5%
Quiet

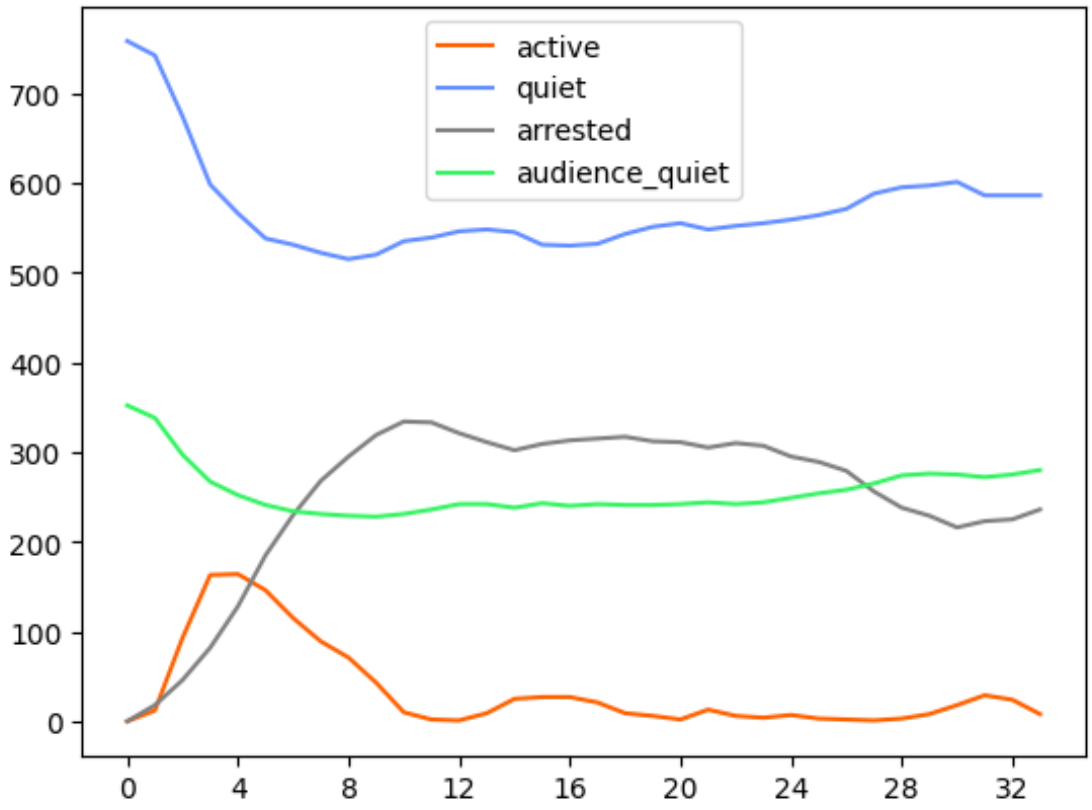
Our early simulation results illustrate how spambots, mod bots, and audience members interact over time within a Twitch-like chatroom environment. The model stabilizes as agents reach equilibrium, where the number of spambots, banned accounts, and quiet audience members remains relatively constant. However, the time to stabilization and the final distribution of agents vary based on parameter settings. Spambot and mod bot vision, mod bot density, and spambot ratio significantly impact the rate of stabilization and the extent of change within the model. When these parameters are increased, the model stabilizes more quickly; when reduced, stabilization takes longer.

For example, keeping all other settings unchanged and adjusting only these parameters results in different model outcomes:

Setting 1:

- Spambot Vision: 8
- Modbot Vision: 8
- Modbot Density: 0.08
- Spambot Ratio: 80%

Higher spambot vision and density accelerate stabilization, increasing active spambots.

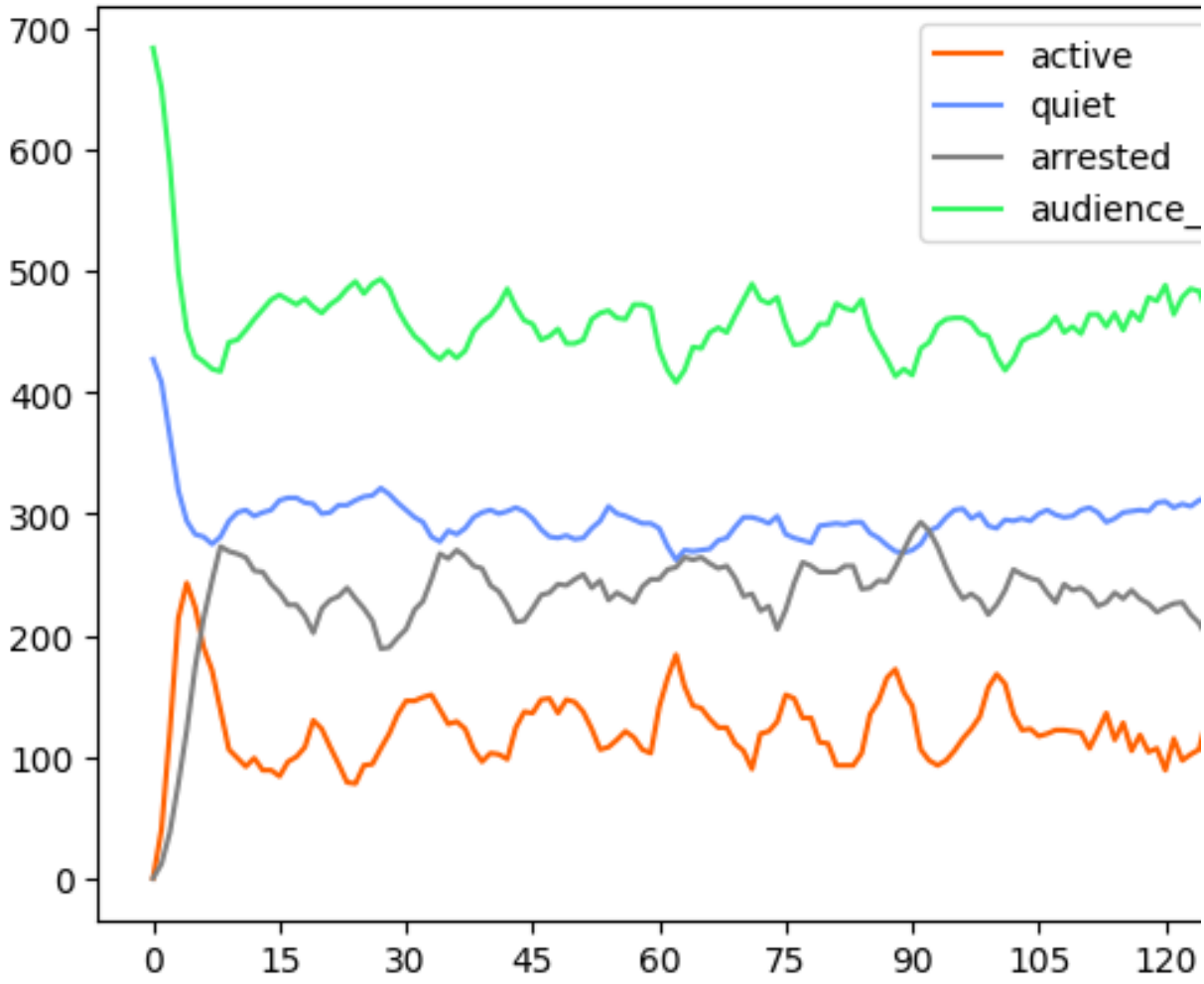


Under the current settings, the model tends to be stable around the 30th step

Setting 2:

- Spambot Vision: 5

- Modbot Vision: 5
- Modbot Density: 0.06
- Spambot Ratio: 60%



Under the current settings, the model stabilizes at about the 120th step, and the fluctuations are large each time

Unexpected Emergent Behaviors

1. **Sudden Spambot Surges**: In some cases, the proportion of active spambots fluctuates sharply, increasing 30-50% within a few steps before stabilizing. This is likely due to synchronized unbanning, where large numbers of spambots re-enter the chat simultaneously.
2. **Chain Reactions**: Audience members, when exposed to spam-dominated chat environments, begin forwarding spam messages, unintentionally contributing to spam proliferation.
3. **Delayed Moderation Impact**: Mod bots require multiple cycles to effectively suppress spam, with effectiveness improving as ban times increase and spam detection accuracy.

Key Findings from Early Runs

- **Registration Difficulty**: Higher registration difficulty reduces the number of active spambots since creating new accounts is more costly.
- **Max Ban Time**: Longer ban times increase the number of banned accounts and reduce active spambots over time.
- **Spambot vs. Mod Bot IQ**: When spambots have a higher IQ than mod bots, they evade detection more effectively, leading to more active spam accounts.

Our experiment revealed unexpected fluctuations after the model stabilized. Initially, we predicted agent numbers would fluctuate within 5%, but instead, active spambots surged by 30-50% within 1-5 steps before returning to equilibrium. This phenomenon likely results from synchronized unbanning, where many spambots are released simultaneously after identical ban durations. This sudden influx triggers a chain reaction; other spambots perceive the environment as safe and resume spamming, while audience members become confused by the surge in misinformation, leading to increased spam activity.

Challenges & Next Steps

Development Challenges

Implementing the simulation presented multiple challenges, particularly in learning the Mesa framework and integrating the Epstein Civil Violence Model to simulate bot interactions. Mesa, while powerful, has a steep learning curve, especially for handling dynamic agent interactions in a grid-based environment. We initially attempted to use SolarViz for visualization, but its complexity led to significant delays, prompting us to rely on simpler visualization tools.

Another challenge was adapting the Epstein Civil Violence Model to accurately represent the Twitch chat moderation ecosystem. Matching agent behaviours such as field of vision, ban difficulty, and intelligence levels—required modifications to ensure realistic interaction dynamics between spambots and mod bots. Initial attempts at designing a detailed intelligence-based spambot hierarchy (e.g., high-IQ and low-IQ spambots as separate classes) proved unfeasible within Mesa's framework, requiring us to simplify our model.

Model Modifications

Due to limitations in Mesa's structure, we revised our approach to managing intelligence levels. Instead of creating separate classes for high-IQ and low-IQ spambots, we treated intelligence as an agent attribute, influencing each spambot's decision-making process. This reduced computational complexity while maintaining meaningful behavioral differences.

Additionally, our initial model lacked a permanent ban mechanism, causing repetitive banning and unbanning cycles. We plan to introduce a ban_forever_time parameter, ensuring that spambots exceeding a set number of bans are permanently removed. This change will enhance the model's ability to reflect long-term moderation strategies.

Planned Refinements for DEL 4.B

To improve the simulation's accuracy and realism, we will focus on the following enhancements:

1. Permanent Ban Mechanism: Implement a system where spambots exceeding a threshold number of bans will be permanently removed, aligning with real-world moderation strategies

on Twitch.

2. Audience Experience System: Introduce a user experience metric where excessive spam exposure or false bans negatively impact audience retention, simulating engagement shifts.
3. Dynamic Learning Mechanism: Enable mod bots and spambots to adjust their IQ dynamically over time, simulating AI adaptation to changing moderation tactics.

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

Attestation

Peter AYADE

- **Conceptualization**: Developed the research focus and overall project goals.
- **Methodology**: Designed the simulation model and aligned it with the Epstein Civil Violence Model.
- **Validation**: Tested and debugged the model to ensure realistic agent behaviours.
- **Writing – Original Draft**: Authored sections on System Overview, Agent Design, and Simulation Environment.
- **Visualization**: Created graphs and charts to represent simulation trends.
- **Writing – Review & Editing**: Edited sections on Data Collection & Visualization and Preliminary Observations.

Mingran CHEN

- **Investigation**: Conducted research on spambot behaviours and Twitch moderation techniques. Data Curation: Managed simulation data collection, tracking emergent behaviours.
- **Software**: Led the development of the Mesa-based simulation, implementing core agent interactions. Validation: Ran parameter tests to analyze bot interactions under different conditions.
- **Writing – Original Draft**: Authored sections on System Overview, Agent Design, and Simulation Environment.

Shenice THOMAS

- **Formal Analysis**: Conducted quantitative analysis of simulation results, identifying trends and anomalies.

- **Investigation**: Researched real-world moderation techniques and their implications for AI governance.
- **Project Administration**: Managed team coordination, documentation updates, and repository maintenance.