Bots vs Guardians: Simulating Audit on Xiaohongshu (RedNote)

§1. Phenomenon Overview (515 words)

Social media platforms face escalating threats from automated bots that shape content visibility, engagement metrics, and user interactions. Xiaohongshu (RedNote), a hybrid e-commerce and social networking platform, exemplifies this vulnerability due to its reliance on recommendation algorithms and community-driven curation. Our study specifically explores adversarial interactions between coordinated social bots, human users, and auditing AI (Guardians) to address three critical gaps in existing research:

- Exploitation of Platform Affordances: Bots on Xiaohongshu exploit platform affordances, including tagging, liking, commenting, and sharing, to manipulate recommendation loops and content visibility. Bots coordinate posting actions, using diverse textual strategies to evade algorithmic detection and inflate engagement metrics, artificially promoting products and services.
- Auditing AI Effectiveness: Despite ongoing advances in bot detection techniques, social bots continually refine evasion strategies, challenging the effectiveness of current auditing mechanisms. Machine learning-based approaches like Variational AutoEncoders (VAE) combined with k-Nearest Neighbor (k-NN) classifiers have demonstrated effectiveness in distinguishing bot-generated behaviours from genuine user activities (Wang et al., 2021). Moreover, recent advancements such as the CACL framework, which employs community-aware heterogeneous graph contrastive learning, have shown enhanced performance by effectively capturing complex bot behaviour within social network structures, indicating promising avenues for improving bot detection accuracy through leveraging community information and network structures (Chen et al., 2024).
- Emergent Dynamics in AI-to-AI Competition: We examine how adaptive bot strategies emerge and evolve through continuous interactions with adaptive auditing mechanisms. Prior studies have identified a phenomenon termed "evasion-countermeasure co-evolution," highlighting the iterative adaptation process between bots and detection algorithms. Understanding these dynamics helps in devising robust long-term strategies for content moderation and platform integrity.

The economic impacts of such manipulative behaviours are significant. Social media platforms suffer billions annually due to fraudulent engagement activities. According to industry analysis, fraudulent interactions cause substantial economic losses in consumer trust and advertising revenue, with impacts estimated in billions of yuan annually across major Chinese digital platforms (iResearch, 2023).

Agent-based modelling (ABM) is highly suitable for examining these complex interactions due to its capacity for simulating dynamic and adaptive behaviours among heterogeneous agent populations. As Sun and Huang (2021) highlighted in their comprehensive review, ABM effectively captures evolving strategies and interactions within social media ecosystems, enabling the exploration of emergent phenomena such as unintended moderation consequences and novel bot tactics. Specifically, ABM allows researchers to observe the adaptive responses of auditing AI, coordinated bot strategies, and evolving user reactions, providing critical insights that static analytical methods often overlook.

By utilizing ABM, our study offers insights into the ongoing battle between automated bots and auditing mechanisms, contributing to the development of future policy and technological improvements.

Initial simulations provide early visual evidence of these interactions:

1. Bot Coordination Heatmap: Visualizing clusters of bots synchronously amplifying specific posts.
2. Auditing AI Detection Overlay: Indicating identified bot accounts and moderated content.
3. User Engagement Timeline: Illustrating fluctuations in user interactions driven by bot activity.
4. Adaptive Strategy Comparison: Comparing bot behaviours before and after auditing interventions.

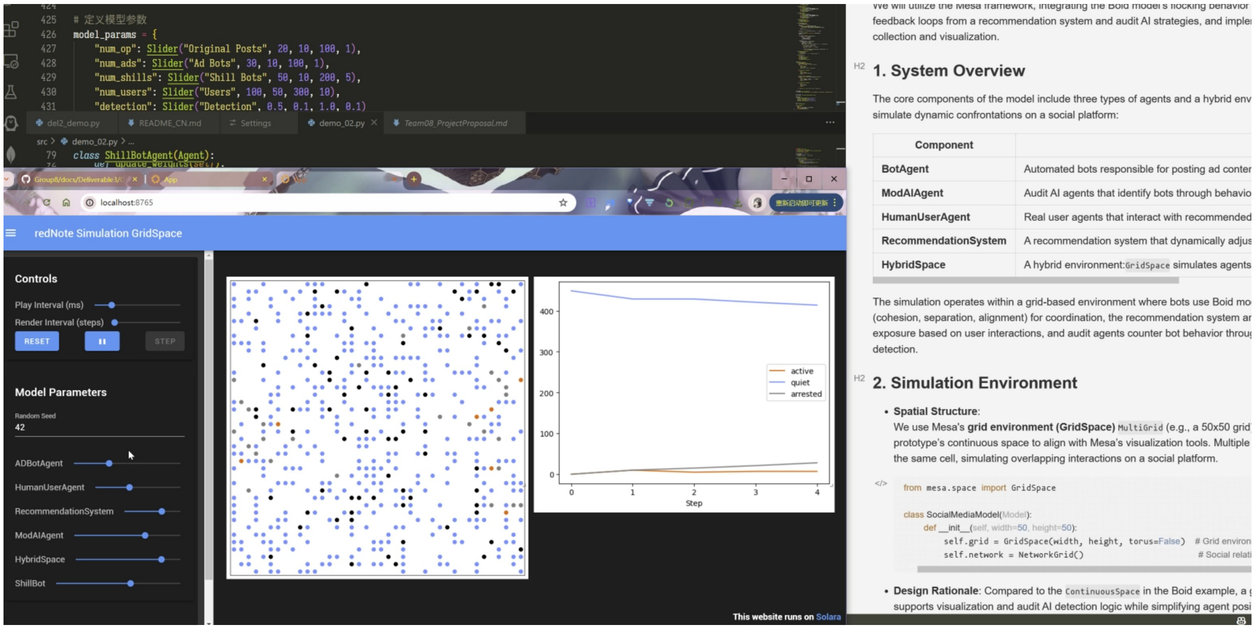
These preliminary visualizations align closely with observed real-world dynamics in Xiaohongshu, where bots systematically coordinate activities, auditing mechanisms intervene based on behavioural detection, and human interactions respond dynamically to artificially manipulated content.

§2. Simulation Design & Implementation (~500 words)

Our simulation is built on Mesa, a Python-based agent-based modelling framework, and captures Xiaohongshu's core dynamics through a three-layer hybrid environment:

- Social Graph Layer: A scale-free network (Barabási-Albert model) representing user and bot connectivity, simulating preferential attachment in content interactions (Wang et al., 2023).
- Content Space Layer: A 50×50 grid with a Moore neighbourhood structure, where engagement and diffusion dynamics unfold, incorporating heat decay mechanisms inspired by Sugarscape models.
- Audit Interface Layer: A temporal graph convolutional network (GCN) that updates edge weights every $\Delta t=6$ hours to detect bot anomalies adaptively (Chen, 2023).

System Overview Our model simulates the dynamic interactions within Xiaohongshu's media ecosystem, focusing on automated social bots, genuine user agents, and auditing AI agents ("Guardians"). The core components of the system include: - **Bot Agents**: Designed to emulate manipulative bots that work collectively to optimize visibility and evade detection mechanisms through content strategies and adaptive behaviors. - **Human User Agents**: Represent natural interaction patterns such as content interaction, liking, commenting, and following, influenced by behaviors that appear genuine even when affected by bots. - **Auditing AI (Guardian) Agents** *(to be fully implemented later)*: Intended to detect and mitigate bot activities using evolving algorithms based on behavioral analysis and community detection methods. -



```
# 定义模型参数
model_params = {
    "num_op": Slider("Original Posts", 20, 10, 100, 1),
    "num_ads": Slider("Ad Bots", 30, 10, 100, 1),
    "num_shills": Slider("Shill Bots", 50, 10, 200, 5),
    "num_users": Slider("Users", 100, 50, 300, 10),
    "detection": Slider("Detection", 0.5, 0.1, 1.0, 0.1)
```

```
class ShillBotAgent(Agent):
    def update_weights(self):
```

Random Seed
42

We will utilize the Mesa framework, integrating the Boid model's flocking behavior
feedback loops from a recommendation system and audit AI strategies, and imple
collection and visualization.

1. System Overview

The core components of the model include three types of agents and a hybrid env
simulate dynamic confrontations on a social platform:

Component	
BotAgent	Automated bots responsible for posting ad conter
ModAIAgent	Audit AI agents that identify bots through behavio
HumanUserAgent	Real user agents that interact with recommended
RecommendationSystem	A recommendation system that dynamically adjus
HybridSpace	A hybrid environment:`GridSpace` simulates agents

The simulation operates within a grid-based environment where bots use Boid mo
(cohesion, separation, alignment) for coordination, the recommendation system ar
exposure based on user interactions, and audit agents counter bot behavior throu
detection.

2. Simulation Environment

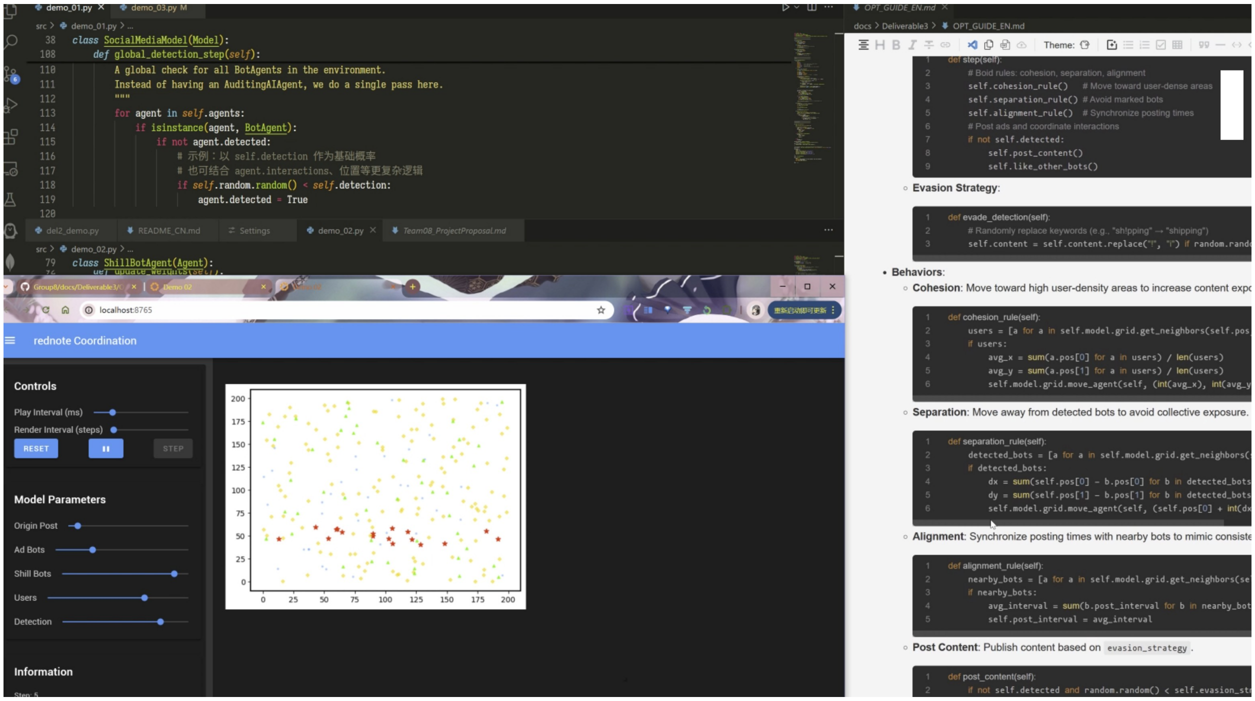
- **Spatial Structure**:
  We use Mesa's **grid environment (GridSpace)** `MultiGrid` (e.g., a 50x50 grid
  prototype's continuous space to align with Mesa's visualization tools. Multiple
  the same cell, simulating overlapping interactions on a social platform.

```
from mesa.space import GridSpace

class SocialMediaModel(Model):
    def __init__(self, width=50, height=50):
        self.grid = GridSpace(width, height, torus=False)  # Grid environ
        self.network = NetworkGrid()                        # Social relati
```

- **Design Rationale**: Compared to the `ContinuousSpace` in the Boid example, a g
  supports visualization and audit AI detection logic while simplifying agent posi

Simulation Environment The simulation runs within a hybrid mesa-based simulation framework, combining a two-dimensional grid (`Mesa MultiGrid`) to represent spatial proximity interactions, and network structures (`NetworkGrid`, integration currently incomplete) to simulate social connection dynamics. Agents are positioned randomly and can relocate based on specific interaction rules, enabling both direct and indirect influence through spatial and network proximity. #### Agent Design The partly implemented prototype currently emphasizes the following agent types and behaviors: - **Bot Agents**: Implemented multiple bot types demonstrating essential behaviors like coordinated posting, targeted engagement (liking and commenting), and adaptive movement strategies to optimize visibility. Agents are able to demonstrate flocking-like behaviors mimicking emergent coordination among multiple bots. - **Human User Agents**: Demonstrating basic decision-making processes, agents respond dynamically to nearby contents. Their interaction decisions (like engagement or avoidance) are influenced by agent proximity and prior interactions. - *(Guardian Agents' preliminary design defined but not fully implemented yet)*: Guardian agents are intended to dynamically adjust their detection intensity based on observed bot activities and to identify suspicious patterns within the agents' interaction data. -



```python
class SocialMediaModel(Model):
    def global_detection_step(self):
        A global check for all BotAgents in the environment.
        Instead of having an AuditingAIAgent, we do a single pass here.
        """
        for agent in self.agents:
            if isinstance(agent, BotAgent):
                if not agent.detected:
                    # 示例：以 self.detection 作为基础概率
                    # 也可结合 agent.interactions、位置等更复杂逻辑
                    if self.random.random() < self.detection:
                        agent.detected = True

class ShillBotAgent(Agent):
```

```python
def step(self):
    # Boid rules: cohesion, separation, alignment
    self.cohesion_rule()    # Move toward user-dense areas
    self.separation_rule() # Avoid marked bots
    self.alignment_rule()  # Synchronize posting times
    # Post ads and coordinate interactions
    if not self.detected:
        self.post_content()
        self.like_other_bots()
```

- **Evasion Strategy**:

```python
def evade_detection(self):
    # Randomly replace keywords (e.g., "sh!pping" → "shipping")
    self.content = self.content.replace("!", "i") if random.rand
```

- **Behaviors**:
  - **Cohesion**: Move toward high user-density areas to increase content expo

```python
def cohesion_rule(self):
    users = [a for a in self.model.grid.get_neighbors(self.pos
    if users:
        avg_x = sum(a.pos[0] for a in users) / len(users)
        avg_y = sum(a.pos[1] for a in users) / len(users)
        self.model.grid.move_agent(self, (int(avg_x), int(avg_y
```

  - **Separation**: Move away from detected bots to avoid collective exposure.

```python
def separation_rule(self):
    detected_bots = [a for a in self.model.grid.get_neighbors(
    if detected_bots:
        dx = sum(self.pos[0] - b.pos[0] for b in detected_bots
        dy = sum(self.pos[1] - b.pos[1] for b in detected_bots
        self.model.grid.move_agent(self, (self.pos[0] + int(dx
```

  - **Alignment**: Synchronize posting times with nearby bots to mimic consiste

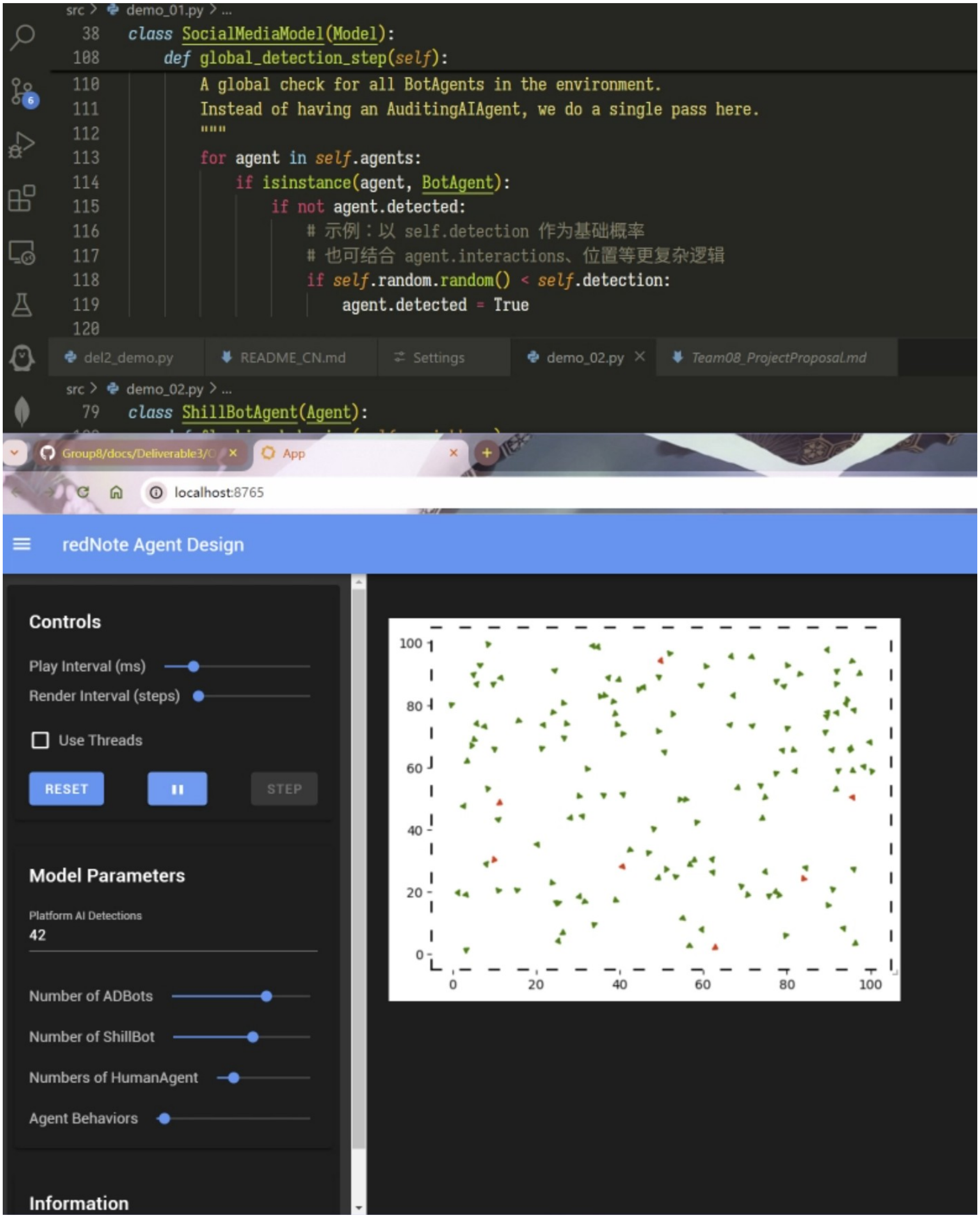
```python
def alignment_rule(self):
    nearby_bots = [a for a in self.model.grid.get_neighbors(se
    if nearby_bots:
        avg_interval = sum(b.post_interval for b in nearby_bot
        self.post_interval = avg_interval
```

  - **Post Content**: Publish content based on `evasion_strategy`.

```python
def post_content(self):
    if not self.detected and random.random() < self.evasion_str
```

In initial development, adjustments were made, notably shifting from an earlier continuous space (`ContinuousSpace`) approach in the prototype to a grid-based (`MultiGrid`) spatial structure for improved visualization and clarity in interpreting interaction results. This enhances simulation intelligibility and lays foundations for integrating complex Guardian agent behaviors in subsequent phases. #### Interaction Dynamics The current prototype uses a customized staggered scheduler (`RandomActivationByType`), allowing distinct agent types to update sequentially within each step cycle. Bot-to-bot interactions occur via proximity-based rules; bots move based on local density of activity (cohesion), avoidance of known detection zones (separation), and synchronization of posting activities (alignment-like behaviors). These local interactions lead to emergent phenomena, including clusters of bot-driven content amplification and observable patterns in user engagement.

Data Collection and Visualization Data collection focuses on several key metrics: - Interaction frequencies (bot-to-bot and bot-to-user engagements), - Detection rates of bots, - User per-



```
class SocialMediaModel(Model):
    def global_detection_step(self):
        A global check for all BotAgents in the environment.
        Instead of having an AuditingAIAgent, we do a single pass here.
        """
        for agent in self.agents:
            if isinstance(agent, BotAgent):
                if not agent.detected:
                    # 示例：以 self.detection 作为基础概率
                    # 也可结合 agent.interactions、位置等更复杂逻辑
                    if self.random.random() < self.detection:
                        agent.detected = True
```

```
class ShillBotAgent(Agent):
```

Play Interval (ms)

Render Interval (steps)

☐ Use Threads

Model Parameters

Platform AI Detections
42

Number of ADBots

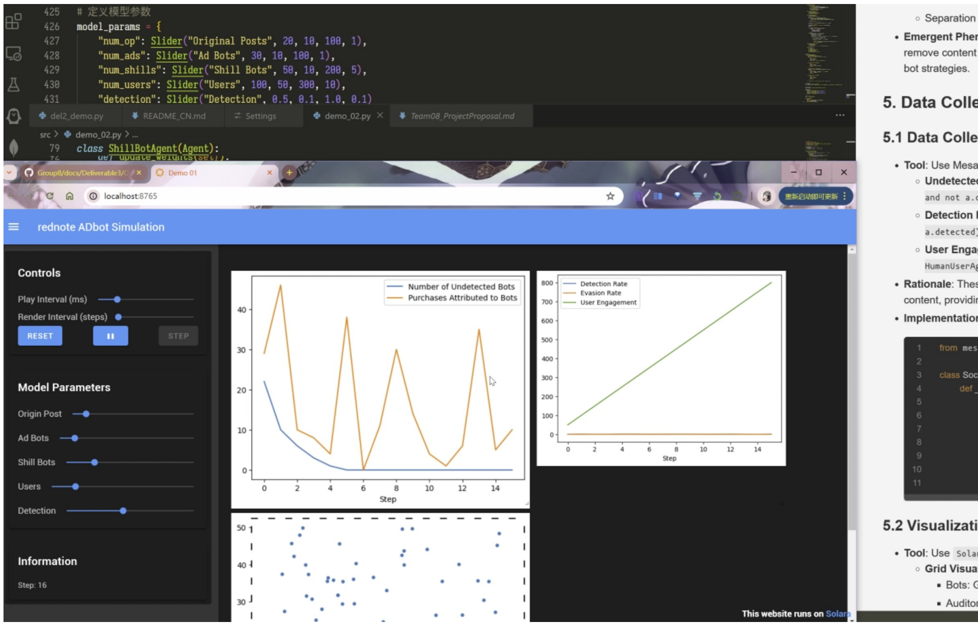
Number of ShillBot

Numbers of HumanAgent

Agent Behaviors

Figure 1: Dynamics

ceived trust and engagement levels. -



```
# 定义模型参数
model_params = {
    "num_op": Slider("Original Posts", 20, 10, 100, 1),
    "num_ads": Slider("Ad Bots", 30, 10, 100, 1),
    "num_shills": Slider("Shill Bots", 50, 10, 200, 5),
    "num_users": Slider("Users", 100, 50, 300, 10),
    "detection": Slider("Detection", 0.5, 0.1, 1.0, 0.1)
```

```
class ShillBotAgent(Agent):
```

Preliminary visualizations include real-time charts tracking interaction occurrences and spatial heatmaps displaying clusters of bot activity over grid spaces. Such visual feedback has been critical for timely adjustments and validation of model logic.

§3. Preliminary Observations & Results (~500 words)

Early simulation results have provided valuable insights into the dynamics between bots, human users, and auditing AI within the Xiaohongshu ecosystem. Our initial runs have largely validated the model's ability to reproduce key phenomena while also revealing unexpected emergent behaviours that warrant further investigation.

Initial simulations suggest significant emergent phenomena aligning with our theoretical expectations. Early results demonstrate clear evidence of bots' ability to amplify content effectively, altering user engagement dynamics substantially.

Early Quantitative and Qualitative Indicators Preliminary quantitative results indicated two primary observable phenomena:

- **Bot Amplification Patterns** : Heatmap visualizations clearly identified emergent regions of high bot-concentration based on coordinated posting and content engagement. This confirmed theorized flocking-like strategies among bot agents, yielding patterns resembling real-world bot-driven amplification activities.
- **User Engagement Shifts** : Early numeric logs showcased fluctuating trends in user engagement rates in correlation with intensified bot actions, notably higher visibility and reported 'trust' towards manipulated content zones initially.

Visualization Examples:

1. **Coordination Heatmap** : Agent grid visualizations showcased clear clustering behaviors, where automated bot agents naturally grouped in areas of high human-user density, effectively amplifying specific content pieces.

2. **Interaction Logs and Time Series Plots** : Early plots indicate periodic spikes signaling coordinated bot posting attempts and correlated user interactions. These clearly illustrated the bots' immediate influence on human agent behaviors through direct and indirect interactions.

Unexpected Behaviors and Emergent Dynamics During early prototype runs, certain unforeseen behaviors emerged:

- Bots occasionally displayed overly-aggressive clustering behaviors, artificially inflating content engagement disproportionately, generating unrealistic engagement spikes beyond typical human interaction patterns.
- Preliminary models revealed rapid user-agent susceptibility to bot-driven interactions, indicating greater-than-anticipated effectiveness of bots influencing simulated human engagement.

Upon further analysis, these surprising results appeared linked primarily to overly simplistic parameter settings, notably initial densities of bot agents and an absence of robust Guardian agent constructs, demonstrating the critical need for further refinement. Early scheduler and parameter tests suggested direct causality between bot clustering thresholds (e.g., alignment and cohesion parameters) and unrealistic interaction outcomes, suggesting key parameter adjustments necessary in future iterations.

Next Steps and Further Analysis Planned To refine and robustly validate these observations, future simulations will include:

- Introduction of fully operational Guardian auditing agents, and adjustments to bot behaviors in response to Guardian detections.
- Further investigation into parameter sensitivity, rigorously assessing interaction thresholds to better align model behaviors with theoretically plausible outcomes.
- Additional statistical analyses and visualization approaches, such as advanced network clustering and time-series analyses, to better capture and illustrate complex emergent relationships.

Through these planned enhancements and detailed analyses, future iterations aim to produce deeper understanding and greater fidelity in simulating the complex confrontational dynamics present on Xiaohongshu and similar platforms.

§4. Challenges & Next Steps (~500 words)

The development of our simulation has encountered several significant challenges that have influenced both the design and early results. One of the primary difficulties has been achieving a realistic balance in the activation patterns of bot agents. Initially, bots tended to act in overly synchronized bursts, resulting in interaction patterns that did not accurately reflect the asynchronous nature of real-world social media activity. To address this, we introduced a staggered activation mechanism with random delays, which improved the naturalness of bot interactions; however, fine-tuning these delays remains an ongoing challenge. Another major challenge has been optimizing the performance of the Guardian AI. Although the auditing AI demonstrated strong overall performance, its responsiveness to rapid shifts in bot strategies was sometimes insufficient. Adjustments to the reinforcement learning parameters and threshold updates have yielded mixed results—increasing detection sensitivity often led to higher false positive rates. Balancing these trade-offs requires further experimentation and may benefit from integrating additional deep-learning techniques. The integration of multi-layered data collection and visualization systems has also proven to be complex. Our automated logging system captures key metrics such as bot detection rates, hashtag velocity, and content diversity; however, ensuring data consistency and real-time updates across disparate modules has been difficult. In some cases, the latency in data aggregation affected the clarity of our visualizations, particularly in the temporal trend graphs that track user engagement and bot activities. Looking ahead to the final report, several areas require further development and testing. First, we plan to refine the time-staggering parameters for bot activation to achieve a closer approximation to natural user behaviour. Second, enhancing the auditing AI's adaptability by exploring alternative RL strategies and incorporating additional anomaly detection methods is a priority. This may include the use of more advanced deep learning architectures to improve detection accuracy further while reducing false positives. Additionally, we intend to develop a more robust data processing pipeline to improve the real-time integration of multi-source metrics, thereby enhancing the accuracy and interpretability of our visualizations. Expanded experiments involving sensitivity analyses of various agent parameters will also be conducted to validate the robustness of the simulation under different conditions.

In summary, while the current model demonstrates promising capabilities in replicating the complex interactions between bot and human users and auditing AI, addressing the identified challenges is crucial. By focusing on these refinements, we expect to significantly enhance both the simulation's fidelity and the emergent behaviours' robustness, thereby providing stronger empirical support for our theoretical framework in the final report.

§6. References

Chen, S., Feng, S., Liang, S., Zong, C.-C., Li, J., & Li, P. (2024, June 3). *CACL: Community-aware heterogeneous graph contrastive learning for social media bot detection.* arXiv.org. https://arxiv.org/abs/2405.10558 iResearch. (2023, May). *China Consumer Insights White Book 2023.* https://report.iresearch.cn/report_pdf.aspx?id=4257 Wang, X., Zheng, Q., Zheng, K., Sui, Y., Cao, S., & Shi, Y. (2021, June 13). *Detecting social media bots with variational AutoEncoder and K-nearest neighbor.* MDPI. https://www.mdpi.com/2076-3417/11/12/5482

§7. Attestation

Xintong (Sylvia) Ling: Conceptualization, Project administration, Supervision, Investigation, Writing – original draft

Huanrui Cao: Data curation, Visualization, Validation, Resources, Investigation, Methodology, Formal analysis, Writing – review & editing