

# What Makes LLM Agent Simulations Useful for Policy? Insights From an Iterative Design Engagement in Emergency Preparedness

YUXUAN LI, School of Computer Science, Carnegie Mellon University, United States

SAUVIK DAS, School of Computer Science, Carnegie Mellon University, United States

HIROKAZU SHIRADO, School of Computer Science, Carnegie Mellon University, United States

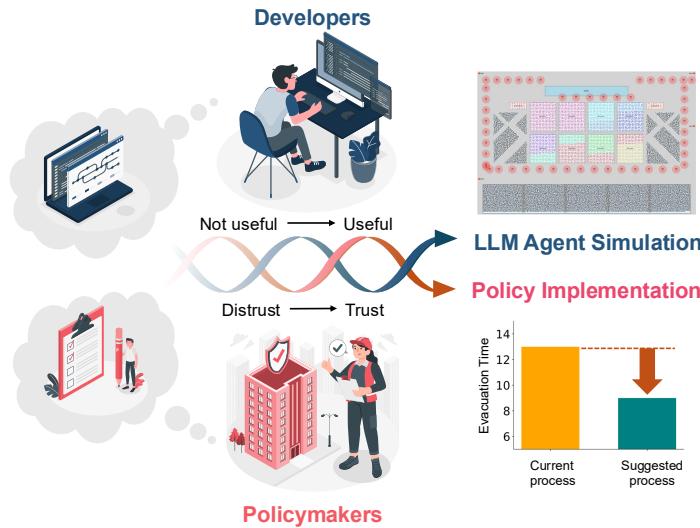


Fig. 1. We report on an iterative design engagement with emergency preparedness policymakers to develop LLM agent simulations that inform policy implementation. Through cycles of design and validation, simulations shifted from being distrusted to trusted, and were ultimately adopted in policy implementation. This usefulness did not follow a linear path from technical sophistication to institutional adaptation, but instead emerged through an iterative, stakeholder-engaged design process.

There is growing interest in using Large Language Models as agents (LLM agents) for social simulations to inform policy, yet real-world adoption remains limited. This paper addresses the question: *How can LLM agent simulations be made genuinely useful for policy?* We report on a year-long iterative design engagement with a university emergency preparedness team. Across multiple design iterations, we iteratively developed a system of 13,000 LLM agents that simulate crowd movement and communication during a large-scale gathering under various emergency scenarios. These simulations informed actual policy implementation, shaping volunteer training, evacuation protocols, and infrastructure planning. Analyzing this process, we identify three design implications: start with verifiable

---

Authors' Contact Information: Yuxuan Li, School of Computer Science, Carnegie Mellon University, Pittsburgh, United States, yuxuanll@andrew.cmu.edu; Sauvik Das, School of Computer Science, Carnegie Mellon University, Pittsburgh, United States, sauvik@cmu.edu; Hirokazu Shirado, School of Computer Science, Carnegie Mellon University, Pittsburgh, United States, shirado@cmu.edu.

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

Manuscript submitted to ACM

scenarios and build trust gradually, use preliminary simulations to elicit tacit knowledge, and treat simulation and policy development as evolving together. These implications highlight actionable pathways to making LLM agent simulations that are genuinely useful for policy.

CCS Concepts: • **Human-centered computing → HCI design and evaluation methods;** • **Computing methodologies → Natural language processing.**

Additional Key Words and Phrases: Social simulation, Policymaking, Iterative design, Large language model, LLM agent

**ACM Reference Format:**

Yuxuan Li, Sauvik Das, and Hirokazu Shirado. 2018. What Makes LLM Agent Simulations *Useful* for Policy? Insights From an Iterative Design Engagement in Emergency Preparedness. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 36 pages. <https://doi.org/XXXXXXX.XXXXXXX>

## 1 Introduction

Large language models (LLMs) have unlocked new possibilities for simulating human behavior [10, 64, 66, 85]. Beyond generating text, LLMs can be deployed as agents, capable of reasoning, conversing, and interacting with defined environments [25, 50, 76]. When deployed at scale, collections of such agents have been shown to reproduce emergent social phenomena, from collective decision-making to information propagation [22, 26, 67]. These findings have inspired a growing thread of research on LLM agent simulation systems, promoted as tools to explore the potential impacts of policies *in silico* before they unfold in practice [39, 45, 47, 51, 52, 56, 57, 74, 83, 88]. Proponents argue that such systems could enable stakeholders to experiment with alternative scenarios and anticipate possible consequences [7, 44, 58, 86].

Yet despite this promise, no policymaking body has, to our knowledge, adopted these systems in decision-making. Instead, they remain largely confined to academic demonstrations, with limited real-world validation and little integration into the workflows, priorities, and institutional constraints of policymakers. Critics caution that LLM agents can risk reproducing stereotypes, amplifying biases, or offering “black-box” outputs that policymakers should not trust [5, 8, 53], while others warn that they may foster false confidence by encouraging decision-makers to treat speculative outputs as predictive truths [46, 89].

These gaps highlight a central tension: while LLM agent simulations hold potential as exploratory tools, their current form falls short of the institutional legitimacy and trustworthiness required for real-world policy use. The tradition of Human-Computer Interaction (HCI) suggests a possible pathway forward, with its rich history of participatory and user-centered design methodologies. These approaches emphasize the involvement of stakeholders in technology development to ensure systems align with actual practices and needs, rather than abstract potential [4, 68, 70].

Building on this tradition, we ask: **How can LLM agent simulations be made genuinely useful for policy?** The “last-mile” usefulness of these simulations can hinge less on advances in models’ capabilities or frameworks’ sophistication and more on the design process itself: engaging policymakers iteratively, grounding simulations in their lived challenges, validating outputs against real-world evidence, and collaboratively exploring how simulation results can inform concrete implementation proposals.

In this paper, we report on a year-long iterative design engagement with a university emergency preparedness and response team—the institutional decision-makers responsible for preparedness policy implementation (hereafter referred to as the “policymakers”). We began with semi-structured interviews to map their existing workflows and decision-making practices, and to probe how simulation might fit within them. We eventually identified graduation commencement—a large, complex gathering event involving thousands of students, families, and staff—as a concrete

policy domain in which evacuation preparedness was both pressing and open to innovation. Over five iterative design phases, we developed and refined an LLM agent simulation system to generate crowd movement and communication dynamics under varying conditions of scale, physical constraints, and information scarcity. At each step, we incorporated policymakers' feedback, validated the simulator against ground-truth data, and worked toward actionable proposals for policy implementation refinement.

Alongside these design iterations, we analyzed our design process to surface broader insights into when and why simulations were perceived as relevant, trustworthy, and actionable by policymakers. Our analysis suggests three key design implications for making LLM agent simulations useful for policy implementation. First, *start with verifiable scenarios and build trust gradually*. Policymakers only engage seriously with simulations when at least part of the output can be checked against reality. Beginning with routine, recognizable cases helps establish credibility, which can then support exploration of more speculative situations. Second, *use preliminary simulations to elicit tacit knowledge*. Even imperfect or “wrong” outputs can be productive: they prompt policymakers to surface the unstated practices and contextual details—such as roles, relationships, or environmental factors—that matter most for policy implementation. Third, *treat simulation and policy development as evolving together*. Simulation capabilities and institutional requirements rarely align from the start. As simulations mature, they shape policymaker expectations, and as policymaker expectations shift, they reshape simulation needs. Framing both as evolving together enables simulations to become durable tools that support decision-making in dynamic institutional contexts shaped by emerging AI capabilities.

Our contributions are threefold:

- (1) **Iterative design of LLM agent simulations in emergency preparedness:** We show how collaborative engagement enabled simulations to evolve from academic prototypes into tools that informed concrete preparedness practices.
- (2) **Insights on simulation-to-policy relevance:** We distill five insights that determine why and how LLM agent simulations are perceived as useful and adopted by policymakers.
- (3) **Design implications for building useful LLM agent simulations for policy implementation:** We provide concrete guidance that developers and designers can apply when developing simulations to inform policy implementation.

Together, these contributions demonstrate not only the potential of LLM agent simulations but also the conditions under which they can move beyond academic demonstration toward institutional uptake.

## 2 Related Work

### 2.1 LLM Agent Simulations and Social Modeling

LLM agents have been shown to produce strikingly human-like behaviors when situated in simulated environments. Early demonstrations revealed that agents equipped with memory and reflection can spontaneously form coalitions, spread invitations, and coordinate collective events [64, 66]. Building on this foundation, researchers have extended the paradigm to diverse domains: clinical workflows in hospitals, classroom dynamics in education, macroeconomic trading, and other large-scale social processes [10, 25, 50, 76, 85]. Multi-agent systems have further been used to explore policy-relevant questions, from epidemic response to online misinformation and environmental decision-making [7, 22, 26, 44, 51, 56–58, 67, 74, 83, 86, 88]. Across these cases, the appeal lies in the promise of “in silico” policy testing—running counterfactual scenarios at scale that would be costly or unethical to attempt in the real world [39, 45, 47, 52].

Yet, despite their rapid technical progress, these simulations remain almost entirely confined to academic demonstration. Evaluation has typically emphasized internal qualities such as coherence or plausibility, rather than external markers of usefulness like institutional adoption or decision-making impact. Policymakers, to our knowledge, have not integrated these systems into practice.

This absence of real-world uptake has fueled a wave of critical concerns. Some argue that LLM simulations risk reproducing stereotypes or producing “black-box” outputs unsuited to accountable governance [5, 8, 53], while others warn that by presenting speculative outputs as seemingly concrete, such systems may foster misplaced confidence in decision-makers [46, 89]. The result is a widening gap: LLM agent simulations grow more sophisticated in technical fidelity, yet remain disconnected from the institutional realities and trust requirements of policy use. Our work aims to address this gap by shifting focus from refining agent architectures to designing processes that make simulations useful in institutional policymaking.

## 2.2 HCI and Policy Implementation

HCI has long been concerned with civic technologies and decision-support systems that shape governance and public policy. Early work introduced platforms for participatory urban planning and neighborhood decision-making [1, 32]. More recent systems like *PolicyCraft* structured deliberations to improve consensus and justification among stakeholders [49]. Studies in crisis informatics highlight similar goals in emergency response, where tools support sensemaking and communication among responders and the public [63, 73, 78].

Across these domains, scholars emphasize the importance of aligning tools with institutional constraints. For example, Corbett and Le Dantec’s ethnographic co-design with a city immigrant affairs office revealed tensions between efficiency-focused digital tools and the relational work officials valued for building trust [9, 20, 21]. Other studies note that legal mandates, organizational hierarchies, and accountability requirements shape how civic or emergency systems can be deployed [6, 9, 69]. While HCI has articulated a vision of technology that makes policymaking more participatory and evidence-driven [28, 54, 79], recent work find that policy actors themselves remain an underrepresented stakeholder in HCI research [87]. This suggests a need for deeper institutional partnerships and design approaches that can bridge between technical innovation and organizational realities [36].

In this study, we build on HCI research that emphasizes aligning technologies with institutional constraints by examining *policy implementation*—defined as the processes through which high-level institutional goals, or *policy* (e.g., protecting life and property in emergencies; see Sec 6.5) are translated into concrete procedures and workflows. Through a year-long partnership with emergency preparedness professionals, we investigate how LLM-agent simulations can support institutional processes, and address the practical realities of policy implementation.

## 2.3 Engaging Stakeholders in Design

To navigate such complexities, HCI researchers have developed a wide repertoire of stakeholder engagement methods. Traditions of participatory design, user-centered design, and community-based co-design have demonstrated how iterative cycles of prototyping, feedback, and negotiation can surface hidden needs and reconcile value tensions [4, 18, 29, 37, 42, 43, 48, 49, 68, 70, 82]. Longitudinal collaborations with municipal offices [21], community organizations [17], and health institutions [62, 72] underscore the importance of partnerships that adapt alongside institutional contexts. Agile and iterative approaches have also been adopted in high-stakes domains such as education, health, and emergency response to foster responsiveness and resilience [3, 71, 77].

With the rise of AI-powered systems, however, new challenges emerge. Unlike traditional tools, AI outputs are probabilistic, opaque, and sometimes unpredictable, complicating stakeholder understanding and co-design [2, 24]. Recent work has explored participatory approaches to algorithm design to surface values, fairness concerns, and accountability requirements [23, 38], while others emphasize new representational tools to make AI legible in design workshops [55, 59]. Together, these efforts point toward a growing consensus: longitudinal, participatory engagements are critical not only for uncovering requirements, but also for fostering trust, negotiating unpredictability, and embedding AI systems into institutional practice. This study contributes to these engagement traditions by addressing the specific challenges of LLM agent simulations for emergency preparedness policy implementation, where outputs are both generative and collective, and where contextual requirements are often tacit.

### 3 Policy Domain: Emergency Preparedness

To explore what makes LLM agent simulations useful for policy, we identified emergency preparedness as our design context. Emergency preparedness represents a uniquely fitting domain because real-world emergency experiments are practically and ethically impossible, observational data from crises are scarce and difficult to generalize, and policymakers must make high-stakes decisions under profound uncertainty [33, 34]. Unlike domains where interventions can be validated through controlled studies, emergency preparedness requires anticipating rare, unpredictable events with limited empirical evidence. LLM agent simulations might offer a way to address these challenges by enabling exploration of evacuation dynamics, testing communication protocols, and examining crowd behaviors under conditions that cannot be safely replicated in reality.

Building on prior work and existing practices, we hypothesized that—given the potential and limitations of state-of-the-art LLMs—the usefulness of LLM agent simulations in policy contexts hinges less on advances in modeling fidelity or computational scale, and more on the design process. Simulations can become valuable when they are developed through iterative engagement with policymakers, grounded in their lived challenges, validated against real-world evidence, and collaboratively explored as tools for shaping concrete policy proposals.

To situate this inquiry, we partnered with a university’s emergency preparedness and response team. This team is responsible for developing and implementing emergency preparedness policies across diverse scenarios—from preparing evacuation procedures for large-scale gatherings such as commencement and orientation, to coordinating responses to natural disasters (e.g., tornadoes, floods), and managing high-stakes emergencies such as active shooter incidents. Because emergencies are inherently rare and information-poor, the team must act under profound uncertainty—balancing institutional policy, safety concerns, and communication clarity while coordinating across stakeholders such as university administration, student groups, local police, and nearby universities.

A central tool in this process is the *after-action report*. After each major university-wide event, the team documents both their preparations beforehand and their actions during the events. These reports highlight improvements over prior processes, identify gaps, and record strategies that worked in practice. The team reviews past reports to refine future policy implementation, the university sometimes adopts successful practices from these reports. In this way, after-action reporting functions as a bridge between day-of response and long-term institutional change, making it a critical site for understanding how simulations might inform policy implementation.

We began our partnership by introducing prior work on model-based emergency coordination and exploring how it could be extended with LLMs. At that stage, the policymaker team already understood both the promise and limitations of social simulations for their domain. They also recognized the natural-language generation capabilities of LLMs as potentially useful for addressing pressing challenges in emergency communication and social media management.

Several members expressed particular interest in generating realistic variations of agent behavior through persona-based prompting techniques, seeing this as a way to capture community diversity and stress-test communication strategies. At the same time, the team made no commitment to incorporate simulation outcomes into their policy implementation, underscoring that the collaboration was exploratory rather than prescriptive.

## 4 Methods

### 4.1 Iterative Design and Participants

We conducted an iterative design engagement with a university’s emergency preparedness and response team between May 2024 and August 2025 – a span of 16 months. This collaboration was structured around flexible meetings convened as opportunities and needs emerged. While the broader team included many stakeholders, our design work primarily involved five core members who were directly responsible for preparedness and crisis communication: a Senior Director for Disaster Recovery and Business Continuity Services (*P1*), a Senior Director for Reputation and Issues Management (*P2*), a Director for Social Media (*P3*), and two Emergency Preparedness and Response Specialists (*P4, P5*). Together, these individuals represented both policy-level decision makers and operational specialists, allowing our process to bridge institutional strategy with on-the-ground practice.

Our design process unfolded in overlapping cycles of needs assessment through interviews, scenario exploration, simulation prototyping, validation, and proposed policy changes based on simulation outputs.

### 4.2 Data Collection and Analysis

We collected multiple forms of data across the year-long engagement, including field notes from design and policy implementation proposal sessions, transcripts of meetings, simulation logs and iteration logs from the system, semi-structured interviews, recordings of the commencement event, and relevant email correspondence. All data collection procedures were approved by the university’s Institutional Review Board (IRB).

We analyzed our qualitative data as follows. The first author transcribed and coded the meeting transcripts, the interviews, the policy implementation proposal session, and relevant email exchanges. Two other researchers independently reviewed the coded transcripts and provided feedback. The first author began by open coding one transcript, then discussed the emerging codes with the other coders to align on coding granularity and analytic focus. We then applied open coding across the full set of transcripts and emails, capturing both immediate design concerns and broader reflections about policy relevance [75]. Together, we refined the codes into higher-level categories and conducted a thematic analysis to surface recurrent patterns [13]. We resolved disagreements through discussion until consensus was reached. These qualitative analyses provided the foundation for the detailed accounts of our design iterations and findings presented in the following sections.

## 5 Design Iterations

We developed our simulation system through multiple iterations throughout the 16-month engagement, with each version incorporating feedback from policymakers and addressing specific design challenges (Fig. 2). In this section, we describe this iterative process that revealed how LLM agent simulations evolve from academic demonstrations with abstract potential to practically useful tools for policy implementation.

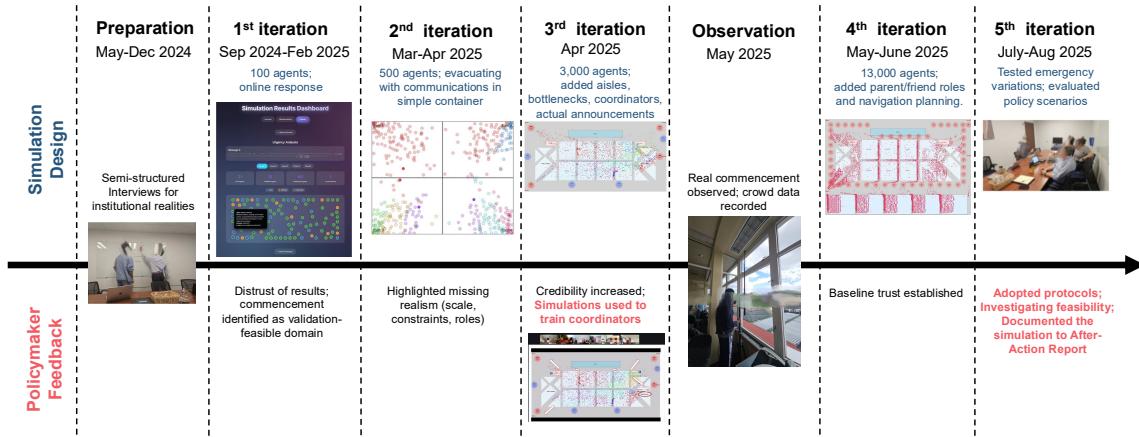


Fig. 2. Iterative design process. Across 16 months (May 2024–Aug 2025), we progressed from preparation interviews through five simulation iterations and an in-situ observation. Each phase introduced greater scale and realism (from 100 to 13,000 agents, adding roles, bottlenecks, and family dynamics) and was shaped by policymakers’ feedback. Early iterations surfaced distrust and missing realism, later iterations built credibility and training value, and by the final iteration, simulations informed adopted protocols, feasibility assessments, and official after-action reports.

### 5.1 Preparation interviews: Capturing Needs and Institutional Realities

To ground and initiate our design process, we conducted three formative interviews during the first five months, complemented by co-creation activities to develop stakeholder and process maps [41, 60].

The first interview, with *P1*, *P2*, and *P3*, followed a semi-structured format and focused on current emergency preparedness practices. We asked them to describe prior experiences, identify challenges in existing workflows, and discuss possible barriers to adopting new technologies. We also invited reflections on possible roles for simulation in their work.

The second interview, conducted with *P1*, examined the policy-making workflow in detail. We asked *P1* to walk us through the sequence of activities involved in developing and implementing emergency policies, grounded in concrete prior experiences. During the session, we co-created a stakeholder map and a process map, documenting actors, responsibilities, and information flows (Fig. A1). After the session, we refined the drafts and invited *P1* to verify their accuracy.

The third interview, conducted with *P3*, investigated daily communication practices. We asked *P3* to describe concrete routines for monitoring and responding to online interactions during emergencies, and probed specific cases where social media shaped people’s perception or altered information exchange.

All three interviews followed a semi-structured format, combining prepared questions with targeted follow-ups to capture both routine activities and situational variations. The finalized stakeholder and process maps are included in the Appendix (Figs. A2, A3, A4, and A5).

### 5.2 First Iteration: Simulating Communication and Misinformation Dynamics

**5.2.1 Simulation Goals and Design.** Based on initial interviews with policymakers, we identified communication challenges as a primary concern during emergencies. *P1*, *P2* and *P3* expressed interest in understanding how official

announcements might be misinterpreted and how misinformation could spread through online social networks during crisis situations.

To approach this, we developed our first prototype simulation, modeling 100 LLM agents interacting “online.” We used GPT-4o to generate message content for each agent, which was prompted to behave as a “student.” In the simulation, agents received lockdown announcements and stay-home orders, then generated responses and posts to share within their social networks. For example, a dormitory restriction notice was interpreted by an anxious agent as a rights violation, triggering an “outrage” response that they shared on a simulated social network. Each agent was given a distinct persona generated through a two-stage prompting process, and operated with a basic state machine to guide decision-making and communication protocols. This structure enabled us to capture both individual variability and emergent patterns in how messages could be (mis)interpreted across a broad population. We implemented two subsystems, focusing on misinterpretation and propagation respectively. See the appendix (Sec A.2) for a more detailed description of the two subsystems. The simulation results showed that variations in the wording of the official announcement influenced how “student” agents exchanged messages and which agents ultimately adhered to instructions.

**5.2.2 Policymaker Feedback.** When we presented this system and its results to *P1*, *P2*, and *P3* in October 2024, they expressed fundamental skepticism about practical utility and real-world validity. Their primary concerns centered on the difficulty of validating simulation outputs against real-world communication dynamics. For example, *P3* questioned whether the model captured realistic communication patterns:

You were showing that message before with the all capital letters... there are all kinds of studies that show that people who are warned about spam emails are warned about emails having an overabundance of capital letters... I think you would find some very interesting differentiations in what the data shows.

The team then pointed out the broader challenge of validation, noting the inaccessibility of complete online communication data and the inherent unpredictability of misinformation incidents. *P1* emphasized the dangers of acting on results that could not be verified:

For example, like if you implement such a simulation result, I can imagine that some high risk of the situation might be so dangerous to use such a simulation result. So maybe for especially the first stage, maybe we come up some application with low risk.

The core challenge centered around validation. Policymakers acknowledged social media misinformation as a pressing concern but emphasized the seeming impossibility of validating such simulations against ground truth data. In response, we explored alternative scenarios that both mattered to the team and offered potential points of observation and verification.

Through these discussions over the following months, we identified evacuation preparedness for the university commencement as a preferable domain. This annual event is the largest gathering on campus, and the team assumed direct responsibility for the safety of the attendees. At the same time, commencement is vulnerable to both natural hazards (e.g., severe weather) and man-made threats (e.g., targeted disruptions), making preparedness a top priority.

While such emergencies are inherently unpredictable, commencement offered a unique opportunity for observation. Because the event recurs yearly, crowd movement and communication patterns can be monitored in real time, providing data against which we might validate simulation outcomes. This made the domain both practically important and methodologically tractable, setting the stage for our second design iteration.

### 5.3 Second Iteration: Simulating “Commencement”

*5.3.1 Simulation Goals and Design.* Building on the lessons from the first iteration, we shifted focus to simulating evacuation scenarios during commencement. Unlike the previous simulation of online interactions, this iteration required modeling both spatial movement and communication among agents.

We simulated 500 LLM agents, again prompted to behave as “students,” and placed them within a rectangular space representing the university stadium where commencement is held. The simulation unfolded over multiple time steps. At the outset, agents received an official evacuation announcement. At each subsequent step, agents exchanged natural-language messages with predetermined peers, simulating online communication with their friends. They then decided which corner of the rectangle to move toward, representing exits from the stadium, with decisions generated through GPT-4o calls.

Each agent was assigned a distinct persona (based on an academic major), though they were identical in terms of physical capabilities and roles. This setup enabled us to explore how communication dynamics and individual interpretations of announcements could influence evacuation flows and crowd movement.

*5.3.2 Policymaker Feedback.* *P1*, *P4*, and *P5* reviewed this iteration in early April 2025. While they appreciated the shift toward a scenario that could in principle be observed and validated, they immediately identified missing elements that undermined the simulation’s credibility and usefulness. For example, *P4* raised practical concerns about scalability that underscored the gap between our simplified model and real-world requirements:

Is your model limited in how many agents it can handle? Commencement can be around 10,000 people, so I’m not sure how the model scales.

Across the discussion, the policymaker team pointed to several absent features: realistic physical constraints (aisles, seating sections, and bottlenecks), physical-body effects that would create realistic congestion patterns, accessibility considerations for diverse populations, the true scale of commencement attendees, the presence of volunteer coordinators that were central to their emergency procedures, and the actual announcement message used. They emphasized that without these contextual details, the simulation felt too abstract to guide concrete policy implementation.

### 5.4 Third Iteration: Simulating the University’s Commencement

*5.4.1 Simulation Goals and Design.* In response to policymakers’ feedback, we incorporated detailed stadium features into the simulation space, including concourses, aisles, and bottlenecks that reflected the actual commencement venue layout. We also increased the number of LLM agents to 3,000. Agents were implemented to adjust their movement speed dynamically based on local congestion, addressing concerns that evacuation safety and efficiency depend on crowd density. We further implemented collision-aware navigation that respected aisle and concourse boundaries while preventing unrealistic oscillations in narrow passages.

To capture variation among attendees, we explicitly introduced heterogeneous agent types. Accessibility considerations were integrated by assigning differentiated mobility parameters to agents with diverse needs. For instance, wheelchair users were placed in designated seating areas and constrained to use specific exits, mirroring actual commencement practices. In addition to “student” agents, we introduced “coordinator” agents positioned at eight fixed locations throughout the stadium. “Coordinator” agents were prompted to guide nearby “student” agents toward specific exits to improve evacuation efficiency, capturing their real-world role in emergency procedures. We consulted policymakers to obtain the official evacuation message and to identify the type of emergency they consider most

important. We then used that evacuation message, as defined by the policymakers, to send to agents during a severe weather emergency—the type of emergency they believe is most likely and most concerning.

This combination of structural realism, heterogeneous mobility, and role differentiation aimed to bring the simulation closer to the practical concerns raised by policymakers.

**5.4.2 Policymaker Feedback.** The response to this iteration marked a significant shift in policymaker engagement. *P1, P4, and P5* reviewed the system at the end of April 2025 and expressed substantially increased confidence in its potential utility. The addition of realistic scale, physical constraints and coordinator roles created what policymakers recognized as a credible representation of their operational environment.

*P1* showed strong interest in the simulation output and spontaneously began envisioning how different coordinator layouts might improve evacuation efficiency, leading to concrete requests for additional scenarios:

Can you run it again until they leave the stadium? Yeah, like the whole process... This helps us figure out where to position volunteers to assist. Right now, we have four volunteers at the bottom of the stadium, but by the time the crowd reaches them, it's too late – they're already bumping into each other. For example, in front of the stage, maybe half could be directed right and half left. If we move the triangles (coordinators) closer to the seating sections, it would help. We may need a red triangle and three purple helper triangles for traffic, since captains alone can't manage that volume... The real inefficiency is when students choose far exits and collide. So rather than placing captains at the exits, it may be better to position them near the sections, as that's when people decide which way to go... This visualization is very helpful for planning locations.

*P4* also immediately identified training applications that we had not originally considered:

This will be also very good for training... because we usually have slides where we explain this is your role, this is what you're going to do, these are the locations, and if we can record a video of a scenario... just so people see this is where you can expect people.

**5.4.3 Policy Implementation.** This iteration marked the first concrete policy implementation. Following their review, policymakers began integrating simulation recordings into their coordinator training sessions before the 2025 commencement. They used visualization outputs to illustrate crowd dynamics and evacuation decision-making processes, replacing static slide presentations with dynamic scenario demonstrations.

Policymakers also documented our simulation approach and findings in their official after-action report following the commencement event. As the university's authoritative record of preparedness practices, the report carries institutional weight by guiding future policies, training, and large-scale safety planning, writing: *This foundational work represents an important resource in understanding crowd dynamics and optimizing emergency response strategies.* The report emphasized the broader institutional value of the research: *This research has been important not only for improving immediate event safety but also for informing future large-scale public safety planning.* Finally, the report committed to sustained interaction of the simulation into preparedness efforts: *The continued development and analysis of this simulation model will contribute to more resilient and adaptive evacuation protocols, ultimately supporting safer environments for all attendees.*

## 5.5 Fourth Iteration: Validating Simulation with Real Commencement

**5.5.1 Empirical Observation.** After the fourth iteration, we had the opportunity to observe the university's commencement alongside the policymaker team in May 2025. In preparation, we examined the physical layout of the venue before Manuscript submitted to ACM

the event to understand how space was actually used during crowd movement. During the event itself, we recorded crowd movement and communication patterns with assuming anonymity under the university's IRB confirmation. We paid particular attention to how attendees exited the stadium once the ceremony concluded. Although no emergency incidents occurred during the event, these recordings served as baseline data for our simulation models.

We also stayed in communication with the team throughout the event. For example, *P1* pointed out several key locations that they were monitoring closely while attendees were leaving the stadium. These insights helped us connect our simulation assumptions with the real points of attention for practitioners, reinforcing the importance of grounding simulations in observable phenomena.

One critical finding was the central role of students' family members. Commencement is not only about celebrating students but also about their significant others. In practice, more than half of attendees were family members. After the event ended, many students immediately sought to find and reunite with their families. As a result, the most severe congestion occurred around family seating areas, where students attempted to reach relatives, which could affect emergency evacuation. Our earlier simulations had included only "student" agents communicating with each other. This observation revealed a major gap, prompting us to reconsider simulation scale, agent roles, and validation points. Family members would need to be modeled as distinct agent types with different goals, movements, and interactions.

**5.5.2 Simulation Goals and Design.** Addressing the gaps identified after the May 2025 commencement observation, we developed the most comprehensive version of our system incorporating the elements policymakers had deemed missing in prior versions. This iteration scaled to nearly 13,000 agents—including both "student" and "family/friends" agents—along with 50 coordinator agents, matching the actual size and structure of the event.

Agents were designed to reflect complex social and spatial behaviors observed in the real world. "Student" and "family/friends" agents were seated in designated areas and paired (or grouped in trios), enabling them to communicate and jointly decide their next destination. Coordinators were positioned throughout the venue to guide nearby groups toward exits, mirroring real emergency procedures.

Figure 3 illustrates the overall architecture of our LLM agent simulator, including both the agent-level decision loop and the stadium-scale interface. See the Appendix (Sec A.3) for a more detailed description of the final version system<sup>1</sup>.

**5.5.3 Policymaker Feedback.** This iteration generated the most comprehensive validation session, held in June 2025. *P1*, *P4*, and *P5* systematically compared simulation outputs against their observations from the recent commencement event and the recordings of the commencement. They confirmed that the system realistically reproduced familiar behavioral patterns and spatial dynamics, marking a substantial step forward in credibility. As shown in Fig. 4, the final simulation closely mirrored crowd dynamics observed during the actual commencement, with similar mobility patterns like congestion near the track areas close to the camera.

*P4* noted how agents' location choices mirrored authentic crowd behavior (emphasis ours):

...there are like agents who choose to stay at the boosters or... stay in the family and friends areas... on the east side and the west side... which is also what we see in the actual commencement, right? Like people around this place and this place... **So this is pretty close reflection because they're talking to each other.**

*P1* emphasized how this baseline validation created the foundation for trusting the system's utility for exploring novel scenarios (emphasis ours):

---

<sup>1</sup>We do not consider the system, itself, to be a core contribution of this work but the design process in which the system was a technology probe.

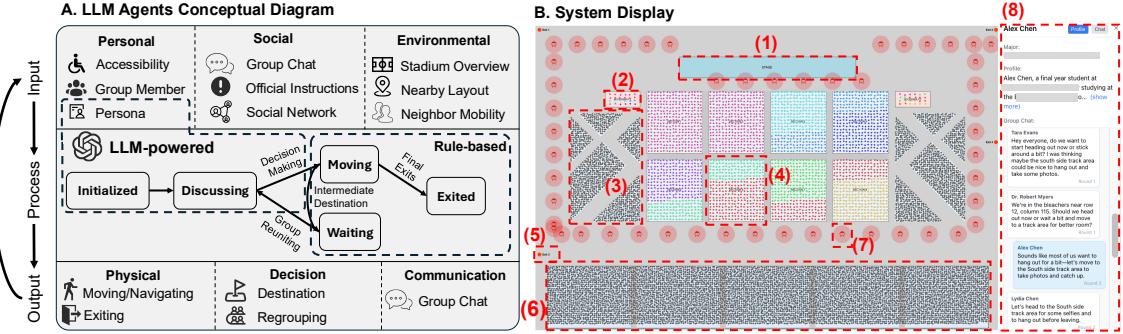


Fig. 3. Simulation overview. (A) LLM Agents Conceptual Diagram: Agents receive inputs from three sources: Personal (e.g., accessibility needs, group membership, persona), Social (group chat, official instructions, social network), and Environmental (stadium overview, local layout, neighbor mobility). For each agent, an LLM governs decision-making and communication and hands off to a rule-based controller for embodied action. Outputs include Physical (moving/navigating, exiting), Decision (destination choice, regrouping), and Communication (ongoing group chat). (B) System display: Stadium-scale interface showing physical layouts, agents as colored dots, coordinators, and a per-agent side panel. Numbered callouts: (1) stage area; (2) students with accessibility needs; (3) seating areas for a portion of students' family and friends; (4) seating sections for students from different majors, each major marked by a distinct color; (5) exits; (6) bleacher area with dense family-and-friends seating; (7) coordinators distributed around the field track for directing flow; (8) a per-agent side panel showing persona attributes including name, major and profile, below are group-chat messages between agents that have social relationships and within the same group. Together this system supports simulations with tens of thousands of agents (up to 13k in our largest runs).

Keep in mind that when we gave you all of those additional feedback pieces and you put it together when you showed a baseline evacuation without anything major, it mirrored the 20 minutes of what you all experienced with us with just the regular evacuation. And **your simulation mirrored what we experience every time we have it. So that gives us some baseline trust**, if you will, to see that it is connecting to what experiences we already had... so that as you all start to make additional trainings, we can take great learnings from it, but we can see how it connects and the pieces make sense as it does that. It's not always throwing out a big red flag of oh my god something is wrong. It's mirroring the stuff that we've seen before as well but then helping us identify additional pieces and so there's a lot of value to that.

This validation success marked another turning point: policymakers began engaging with the system not only to verify known outcomes but also to explore hypothetical emergency scenarios. They spontaneously identified policy implementation opportunities, particularly around coordinator positioning strategies and differentiated announcement messages for specific emergency types. They also expressed interest in testing how variations in coordinator layouts might affect evacuation efficiency and how tailored communication protocols could improve response effectiveness.

## 5.6 Fifth Iteration: Exploring Better Policy Implementation

**5.6.1 Simulation Goals and Design.** For this iteration, we used the validated system to explore simulation outputs for novel scenarios and policy implementation improvements identified by policymakers. Building directly on their feedback, we implemented a set of scenario variations that combined:

- two emergency types (severe weather and bomb threat)
- evacuation announcements with and without specific location information,
- alternative coordinator positioning strategies

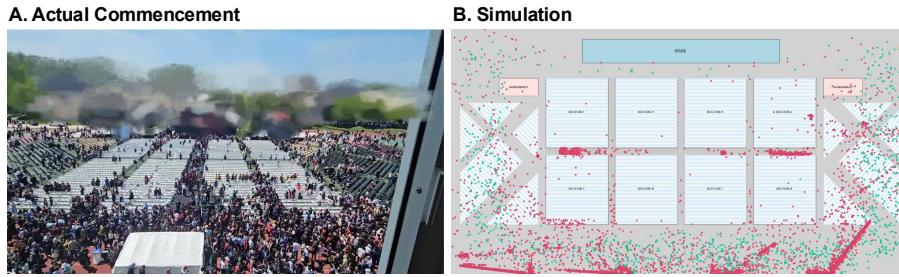


Fig. 4. Comparison of crowd movement dynamics between empirical record and simulation. (A) Photograph of the 2025 commencement, showing attendee movement as the ceremony concluded. (B) Simulation snapshot of the same scenario, incorporating 13,000 agents with roles for students, family members, and coordinators, represented as colored dots. The simulation reproduced key spatial patterns observed in the real event, such as congestion on the track areas at the bottom of the frame, providing a credible basis for validating the system.

- the addition of an extra exit

We tested severe weather scenarios where all exits remained accessible and bomb threat scenarios where specific areas required targeted evacuations away from threat locations. Announcement variations included location-specific messages informing agents about threat locations versus general evacuation instructions. Coordinator layouts compared existing positioning with alternative distributions such as placing coordinators closer to seating sections rather than concentrated near exits.

These variations were designed to reflect the specific “what-if” questions raised by the policymaker team and to test the robustness of different preparedness strategies. To evaluate the impact of these policy implementation options, we measured evacuation efficiency by the time required for 80% of agents to complete their evacuation—the outcome metric that policymakers consistently emphasized as most relevant for assessing operational success.

**5.6.2 Policy Implementation.** This iteration in August 2025 produced three concrete policy implementation proposals that emerged directly from simulation findings:

- (1) Opening a previously unused exit at the northwestern corner of the stadium to reduce evacuation bottlenecks
- (2) Training coordinators to guide people located both near to and far away from localized threats rather than focusing only on nearby individuals
- (3) Developing differentiated evacuation protocols for various emergency types and incorporating these distinctions into coordinator training materials.

The policymaker team adopted Proposals 2 and 3, incorporating the expanded coordinator guidance protocols and differentiated emergency procedures into their standard operating protocols. However, they also noted that in man-made threat scenarios—such as bomb or active-shooter incidents—operational authority shifts to local police. In such cases, the preparedness team’s role would be limited, constraining their ability to fully implement these process changes. For Proposal 1, policymakers initiated a formal feasibility assessment process to evaluate the structural and logistical requirements for opening the additional exit, representing the standard institutional pathway for implementing infrastructure changes.

The after-action report in August 2025 documented these outcomes, emphasizing both sustained institutional integration of the simulations and their broader significance for large-scale public safety planning. Policymakers explicitly committed to ongoing development of the system as a long-term resource for preparedness and training.

## 6 Findings

In our study, “usefulness” did not follow a linear progression from technical sophistication to institutional adaptation. Rather, it emerged through an iterative, stakeholder-engaged design process in which policymakers gradually built trust in simulation outputs, learned the boundaries of what LLM agent simulations can and cannot do, and re-calibrated their expectations and criteria based on this hard-won trust and understanding.

Analyzing this process, we identified five key insights on developing LLM agent simulations that are “useful” for policy by supporting implementation:

- (1) **Validation Filter:** Usefulness requires scenarios that can be validated
- (2) **Trust Bootstrap:** Trust from validating mundane scenarios extends to novel ones
- (3) **“Fix-It” Response:** “Wrong” simulations surface tacit domain knowledge from policymakers
- (4) **The Details Matter:** Contextual nuance in interaction environments enables policy use
- (5) **Policy-AI Interaction:** Usefulness emerges from co-evolution in policy requirements and simulation capabilities

We discuss each of these insights in the subsections that follow.

### 6.1 Validation Filter: Usefulness Requires Scenarios That Can be Validated

In our study, LLM agent simulations were not deemed *useful* in general, but only in scenarios where outputs could be *validated* against policymakers’ prior experience or ground-truth data. Policymakers did not state this requirement explicitly at the outset, but over time it became clear that observable, verifiable scenarios were essential entry points for building confidence in simulation systems. Only after this baseline validation did they begin to engage with outputs for less well-understood or hypothetical emergency scenarios.

We describe this as a “validation filter”: LLM agent simulations can be perceived as useful only in application domains where results from well-understood scenarios can be checked by multiple stakeholders together. This filter is particularly important given the stochastic nature of LLM outputs: both policymakers and developers needed to see where the system aligned with known realities before exploring how it might generate value in untested situations.

This observation emerged immediately in our first iteration, which simulated social media misinformation dynamics. Despite acknowledging misinformation as a pressing concern, policymakers expressed fundamental skepticism about validation possibilities. As *P3* explained:

It’s very hard for us to access social media data in previous emergencies, and some important communication may happen at private Discord channels... And then you have people in other countries who may be having a conversation that’s blowing up way beyond what is actually happening here... we don’t know, I mean, I don’t know how to communicate directly with some of our audiences overseas.

Because of this gap, the domain was deemed unsuitable for useful simulation, despite its clear policy relevance. Policymakers instead steered discussions toward scenarios with verifiable outcomes, explicitly welcoming our proposal to observe commencement. As *P1* put it:

We will be more than happy to have you here and so what we do is what we actually have an emergency operation center... you have the whole view and you can see actually see the flow of people.

## 6.2 Trust Bootstrap: Trust from Validating Mundane Scenarios Extends to Novel Ones

The validation filter creates a paradox: Simulations are most valuable for situations that are difficult to observe or anticipate [10, 64, 65], yet they can only be trusted if grounded in scenarios that are already well understood. We found that this paradox could be addressed through a “trust bootstrap” process.

In our study, once trust was established through successful validation in mundane scenarios (in our case, attendees exiting the stadium after commencement), that trust could be extended—or “bootstrapped”—to more novel and speculative scenarios where robust validation was harder or practically impossible. This process has become especially salient with recent advances in LLMs, which allow simulation systems to adapt to natural-language scenarios while building on validated setups.

This dynamics was reflected in *P1*'s comment during the fourth iteration:

...when you showed a baseline evacuation without anything major, it mirrored the 20 minutes of what you all experienced with us with just the regular evacuation. And your simulation mirrored what we experience every time we have it. So that gives us some baseline trust, if you will, to see that it is connecting to what experiences we already had.

This baseline trust enabled policymakers to explore more speculative possibilities. For example, in our fifth iteration, we demonstrated that opening an additional exit at the northwestern corner could speed up evacuation. In response, *P5* proposed another option:

What if there was another exit that could potentially be opened?... Right behind it are stairs that lead up to the buildings. I believe the university runs a fence across the back because it's behind the stadium. Maybe the university could put a temporary gate piece there instead of fencing it off completely... The model (simulation) would be good for this and this is kind of the next step on the maturing the process and considering those additional options.

We tested the behind-stage exit, but the simulation outputs showed it was less efficient than the northwestern exit. As a result, the team did not pursue it further. This underscores how validated trust allowed policymakers to both imagine new options and critically assess their feasibility.

## 6.3 “Fix-It” Response: “Wrong” Simulations Surface Tacit Domain Knowledge from Policymakers

We also found that iterative process was essential for developing trust because it allowed policymakers to respond to simulations with “fix-it” reactions. Even when policymakers were deeply familiar with their domain, they did not initially know which details were critical for interpreting a simulation until they saw something that looked off. Incomplete, simplistic, or otherwise “wrong” simulations therefore served as productive technology probes [42]. By reacting to outputs that seemed unrealistic or missing, policymakers revealed hidden assumptions and unspoken expertise, which in turn guided refinements that brought the simulations closer to institutional relevance.

For example, our original commencement evacuation simulation did not consider people’s accessibility situations. When observing that LLM agents move in identical speed smoothly, *P5* immediately flagged this omission:

Did you think about like people with disabilities and they might want to exit like through the one that has like the ramp or anything like that? Did you guys think about those factors?

This requirement—accounting for accessibility situations—did not emerge until after policymakers saw a simulation that failed to include it. As *P1* later reflected:

We are aware that there's so many variables, so we're not going to be going to be able to simulate 100% the whole. So when we trust the system enough... I think that to mutually learn... Because I think this is for us this is a learning experience. So we know that the system is giving us baseline data, but it's also as good as the information that we provide... to build an own understanding of how the process works.

In this way, unrealistic or incomplete outputs did not simply highlight errors—they revealed hidden assumptions about crowd psychology and institutional protocols. These insights were not captured through direct questioning but surfaced only when policymakers engaged with “wrong” simulations.

#### 6.4 The Details Matter: Contextual Nuance in Interaction Environments Enables Policy Use

LLM agent simulations consist of two core elements: the agents themselves and the environments in which they interact [76]. In our early iterations, we focused primarily on designing agents and leveraging LLM capabilities. However, policymakers consistently pointed out that without sufficient nuance in the interaction environment, the simulations lacked credibility.

In our case for commencement evacuation, these contextual details included the physical (stadium layout, seating), social (family relationships, accessibility needs), procedural (coordinator roles, protocols), and temporal (event sequences, decision timelines). When these contexts were incorporated, policymakers not only trusted the realism of the outputs but also began to envision practical uses. For example, in the third iteration, seeing weather-related announcements integrated into the simulation prompted *P4* to imagine how the same system could handle more serious threats:

We were thinking that this evacuation is due to severe weather, which would be a fairly benign reason.

But it could also be because of a bomb threat, and that opens up a whole different conversation.

In academic demonstrations, developers often concentrate on designing LLM agents. However, our findings show that the interaction environment matters just as much. Context reinforced utility by anchoring simulations in recognizable realities, and it sparked imagination about new applications. Missing or misaligned contexts could cause temporary dips in confidence, but these gaps also triggered “fix-it” responses that revealed additional requirements (see Sec. 6.3).

#### 6.5 Policy-AI Interaction: Usefulness Emerges From Co-Evolution in Policy Requirements and Simulation Capabilities

Our iterative process ultimately revealed a coevolution between policy requirements and simulation capabilities. While we initially aimed to design simulations that could directly inform “policy,” policymakers clarified that our assumption was misplaced. *P1* stated:

These are process changes, not policy or concept of policy is different. Our policy is to implement safe procedures to make sure that we protect life, property. So, policy has a context of what we intend to do... So, our policy is the same. What we’re trying to do is to find ways to fulfill the policy, which is protecting life.

This distinction—between stable, high-level policy commitments and the evolving processes used to implement them—proved crucial for understanding where simulations could have impact. Policy represents overarching commitments that remain constant across scenarios, while policy implementation, such as processes and procedures, changes in response to context.

Usefulness therefore emerges recursively. Simulations are not one-directional tools for generating new policies; rather, they become valuable when policymakers use them to refine how existing policy goals can be operationalized

in practice. In this way, both problems and solutions evolve iteratively alongside simulation development, creating a feedback loop between simulation design and policy implementation.

## 7 Discussion

### 7.1 Are LLM Agent Simulations Useful for Policy?

Scholars remain divided on the usefulness of LLM agent simulations. This debate reflects a fundamental tension captured in Bill Buxton’s distinction between “*building the thing right*” versus “*building the right thing* [16].” Proponents emphasize “*building the thing right*,” focusing on technical sophistication and agent architectures [39, 47, 51, 64, 66, 88]. Critics, by contrast, question whether such simulations are the “*right thing*” at all, warning against the risks of treating speculative outputs as predictive truths [5, 8, 46, 53, 89].

Our work suggests a third path: through iterative, stakeholder-engaged design, LLM agent simulations can *become* the right thing within particular policy contexts. Rather than assuming simulations are inherently useful or inherently useless, we found that usefulness emerges through a design process in which developers and policymakers jointly explore a design space defined by the intersection of technical possibilities and policy needs.

Indeed, by situating simulation development within the everyday decision-making practices of emergency preparedness professionals, we observed how LLM agent simulations could have practical utility. The simulations did not replace policymaker judgment or provide absolute predictive forecasts. Instead, they functioned as technology probes, surfacing tacit knowledge of the problem domain and enabling the testing of hypotheses that would be costly, or impossible to trial in the real world. Policymakers gradually progressed from validating simulations against mundane scenarios to using them for novel explorations, converting abstract outputs into concrete training protocols and evacuation procedures.

At the same time, caution is warranted. As trust in simulation outputs builds, there is a risk that model errors and biases risk will be overlooked. In particular, demographic biases in agent behavior present a two-sided challenge [53]. On the one hand, such biases can sometimes be intentionally leveraged to mirror real-world disparities and stress-test policies against them. On the other hand, uncritical reliance on these biased outputs—especially in policy decision-making—can reinforce inequities rather than mitigate them [61]. Developers and policymakers must therefore carefully distinguish between *accepting bias to appropriately model reality* and *allowing bias to shape policy recommendations*.

The significance of our work lies not in claiming that LLM agent simulations should universally guide policy implementation, nor that iterative design automatically guarantees usefulness. Instead, our study demonstrates how LLM agent simulations can become useful under specific design conditions. The central question is not *whether* such simulations are useful for policy, but *how we can design them to be useful for policy*. Our case demonstrates one such pathway: not inevitable or universally generalizable, but achievable through iterative, context-sensitive design.

### 7.2 How Can We Design LLM Agent Simulation That Are Useful for Policy?

While our findings build on standard HCI practice, that iterative stakeholder engagement maximizes our chances of making useful technology, this section highlights distinctive design conditions specific to LLM agent simulations for policy use. These conditions double as design implications, outlining how such simulations can be developed and deployed in ways that are genuinely useful for policy by supporting implementation. They are intended to guide developers, designers, and policymakers working at the intersection of HCI, AI, and public decision-making.

**7.2.1 Start with Verifiable Scenarios and Build Trust Gradually.** Unlike traditional tools, LLM agent simulations claim to model complex social phenomena that cannot be directly verified [26, 40, 47]. However, our “validation filter” (Sec 6.1) finding shows that simulations only become useful when some baseline of validation against observable reality is possible. Policymakers are unlikely to accept simulation outputs they cannot check, since LLM agent simulations produce outputs that are inherently unpredictable. Instead, trust is built post-hoc, when simulated outcomes align with familiar or observable patterns.

This unpredictability is also the strength of LLM agent simulations—they allow us to explore unanticipated situations through agent interaction—but it makes the question of validation especially challenging. To address this challenge, developers and designers should first identify verifiable cases that can serve as foundations [35] and deliberately stage this progression: begin with routine scenarios stakeholders recognize, confirm alignment with their expectations, and only then introduce speculative cases. This “trust bootstrapping” approach (Sec 6.2) allows LLM agent simulations to harness their generative capacity while extending policymakers’ confidence from validated setups to novel, unverified scenarios.

**7.2.2 Use Preliminary Simulations to Elicit Tacit Knowledge.** In HCI, prototyping is not just about refining and reducing errors, but about exploring ideas through progressive increases in fidelity as designers gain confidence [15, 30, 65, 90]. LLM agent simulations for policy implementation can benefit from the process in similar ways, but our “fix-it” response finding (Sec 6.3) highlights an additional benefit. With the verisimilitude of LLM-generated behaviors, even deliberately preliminary or “wrong” simulations can generate the most valuable conversations with policymakers, surfacing tacit domain knowledge that otherwise they cannot articulate upfront.

This marks a clear departure from typical AI development processes, which emphasize system accuracy and polished final outputs [14, 19]. All simulations are necessarily imperfect—they cannot capture an entire world. In fact, as Bonini’s paradox reminds us, models can become *less* useful as they attempt to capture more of a complex system [11]. Like a map that is most useful when it selectively simplifies the territory, the challenge in simulation design is not simply to increase fidelity, but to choose carefully which aspects to represent in order to make the model actionable.

Developers and designers should therefore treat interim simulations not just as flawed prototypes to be corrected, but as probes that invite critique. Framing outputs with questions such as “*what’s missing here?*” rather than “*is this correct?*” helps uncover tacit domain knowledge and institutional practices that can ultimately make the system more useful. Importantly, the missing elements are not always tied to LLM capabilities themselves. Policymakers care about the broader social context, beyond agent behavior alone. In our case, much of the feedback centered on the interaction environment—such as spatial layout, family relationships, or coordinator roles—which ultimately proved crucial for making the simulations useful for policy and its implementation (Sec 6.4). These findings underscore that the details surfaced through iterative feedback are not incidental: they are precisely what determines whether a simulation achieves the fidelity needed to inform policy implementation.

**7.2.3 Set a Design Focus on Evolving Simulation Capabilities and Policy Requirements Together.** User-centered design typically assumes relatively stable user needs that technology should accommodate [4, 27, 80]. However, we found that simulation capabilities and policymaker requirements were inter-dependent and developed simultaneously, with neither existing in completing form at the outset (Sec 6.5). In our case, simulations began as simple demonstrations of evacuation patterns, but over time policymakers used them to experiment with coordinator layout and differentiated protocols. At the same time, several suggestions based on our simulations were not pursued—even after trust in the system was established—because of institutional constraints. One example was for local law enforcement to assume

authority during bomb threats. This contrast shows that simulation and policy implementation development can only move forward together when technical possibilities and institutional boundaries are both taken into account. For complex AI systems in institutional contexts, precise design goals and useful solutions cannot be fully defined at the onset, but must be iteratively co-constructed.

Developers and designers should therefore set a design focus for evolving both simulation capabilities and policy requirements as interdependent targets, rather than treating one as subordinate to the other. This means focusing design on decision spaces where adaptation is possible—such as resource allocation, training, or communication—while building flexibility so that both the system and institutional processes can adjust as shared understanding deepens. By embedding this principle of interaction into the dual design focus from the outset, simulations can move beyond prototypes to become enduring tools for practice.

### 7.3 Limitations and Future Directions

Our single-domain focus limits generalizability. Emergency preparedness offers clear success metrics and observable validation opportunities that may not exist in other policy domains. The close collaborative relationships with policymakers may also have influenced outcomes in ways that differ from typical technology adoption scenarios.

Resource requirements raise further questions of scalability. Our iterative design process required substantial time and commitment from both developers and policymakers—resources many institutions may lack. User-centered automatic interpretation techniques could help increase both the interpretability and scalability of simulation results [31, 81, 84].

Ultimately, the usefulness of such tools should be judged by whether suggested changes in policy implementation actually improve safety during emergencies. Our study did not directly test implementation results. Longitudinal studies are needed to evaluate this question and to trace how simulation tools evolve over extended periods of institutional use. Future research should also test these design conditions across other policy domains and organizational contexts, and explore ways to scale iterative methods to institutions with fewer resources. Developing evaluation frameworks that center stakeholder experience, rather than traditional accuracy alone, represents another important methodological direction.

Our findings are also constrained by the current capabilities of LLMs—we primarily used OpenAI’s GPT-4o and GPT-4.1. As these models advance, new opportunities and challenges for simulation design will likely emerge. Nonetheless, our study suggests that building useful LLM agent simulation systems requires shifting focus from technical sophistication toward institutional alignment. The path from technological demonstration to institutional impact runs not only through better models, but through deep engagement with the organizational and political logics of policymaking itself.

## 8 Conclusion

In this paper, we presented a year-long iterative design of an LLM agent simulation system with a university emergency preparedness team, showing how participatory engagement enabled simulations to move from academic demonstrations to institutionally integrated tools. Our findings demonstrate that usefulness arose not from technical fidelity alone, but from design practices that helped build trust and utility gradually. By tracing this pathway, we conclude with design implications for practitioners at the intersection of HCI, AI, and policymaking: begin with validation-feasible domains, treat imperfections as opportunities to elicit expertise, and set a design focus on simulation and policy implementation together.

As famously observed in statistics, “*All models are wrong, but some are useful*” [12]. Our study examines how LLM agent simulations can be designed to be useful for policy, not by claiming predictive certainty, but by aligning technical

possibilities with institutional needs. This work provides a stepping stone for harnessing emerging AI capabilities in service of human well-being through more effective policy implementation.

## 9 Acknowledgments

Portions of the teaser figure (Fig. 1) are adapted from illustrations by Storyset (<https://storyset.com/technology>; <https://storyset.com/work>). We thank Q. Xiao for his insightful advice on the qualitative analysis. This work was supported by the NOMIS foundation and the National Science Foundation under grant #2316768.

## References

- [1] Sahar S Abdalla, Sarah A Elariane, and Sarah H El Defrawi. 2016. Decision-making tool for participatory urban planning and development: Residents' preferences of their built environment. *Journal of Urban Planning and Development* 142, 1 (2016), 04015011.
- [2] Ashraf Abdul, Jo Vermeulen, Danding Wang, Brian Y Lim, and Mohan Kankanhalli. 2018. Trends and trajectories for explainable, accountable and intelligible systems: An hci research agenda. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–18.
- [3] Pekka Abrahamsson, Outi Salo, Jussi Ronkainen, and Juhani Warsta. 2017. Agile software development methods: Review and analysis. *arXiv preprint arXiv:1709.08439* (2017).
- [4] Chadie Abras, Diana Maloney-Krichmar, Jenny Preece, et al. 2004. User-centered design. *Bainbridge, W Encyclopedia of Human-Computer Interaction*. Thousand Oaks: Sage Publications 37, 4 (2004), 445–456.
- [5] William Agnew, A Stevie Bergman, Jennifer Chien, Mark Diaz, Seliem El-Sayed, Jaylen Pittman, Shakir Mohamed, and Kevin R McKee. 2024. The illusion of artificial inclusion. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–12.
- [6] Kars Alfrink, Janus Keller, Neelke Doorn, and Gerd Kortuem. 2023. Contestable camera cars: a speculative design exploration of public AI that is open and responsive to dispute. In *Proceedings of the 2023 CHI conference on human factors in computing systems*. 1–16.
- [7] Jacy Reese Anthis, Ryan Liu, Sean M Richardson, Austin C Kozlowski, Bernard Koch, James Evans, Erik Brynjolfsson, and Michael Bernstein. 2025. LIm social simulations are a promising research method. *arXiv preprint arXiv:2504.02234* (2025).
- [8] Sherry R Arnstein. 1969. A ladder of citizen participation. *Journal of the American Institute of planners* 35, 4 (1969), 216–224.
- [9] Reuben Binns, Max Van Kleek, Michael Veale, Ulrik Lyngs, Jun Zhao, and Nigel Shadbolt. 2018. 'It's Reducing a Human Being to a Percentage' Perceptions of Justice in Algorithmic Decisions. In *Proceedings of the 2018 Chi conference on human factors in computing systems*. 1–14.
- [10] Marcel Binz, Elif Akata, Matthias Bethge, Franziska Brändle, Fred Callaway, Julian Coda-Forno, Peter Dayan, Can Demircan, Maria K Eckstein, Noémi Éltető, et al. 2025. A foundation model to predict and capture human cognition. *Nature* (2025), 1–8.
- [11] Charles P Bonini et al. 1967. *Simulation of information and decision systems in the firm*. Markham Chicago.
- [12] George EP Box. 1976. Science and statistics. *J. Amer. Statist. Assoc.* 71, 356 (1976), 791–799.
- [13] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.
- [14] Miles Brundage, Shahar Avin, Jasmine Wang, Haydn Belfield, Gretchen Krueger, Gillian Hadfield, Heidy Khlaaf, Jingying Yang, Helen Toner, Ruth Fong, et al. 2020. Toward trustworthy AI development: mechanisms for supporting verifiable claims. *arXiv preprint arXiv:2004.07213* (2020).
- [15] Marion Buchenau and Jane Fulton Suri. 2000. Experience prototyping. In *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*. 424–433.
- [16] Bill Buxton. 2010. *Sketching user experiences: getting the design right and the right design*. Morgan kaufmann.
- [17] Mirian Calvo and Annalinda De Rosa. 2017. Design for social sustainability. A reflection on the role of the physical realm in facilitating community co-design. *The Design Journal* 20, sup1 (2017), S1705–S1724.
- [18] Lingyun Chen, Zitao Zhang, Muwu Shan, Qing Xiao, and Eli Blevis. 2025. Sustainable Robot Future: A Speculative Design about Humanity, Robots, and Ecology. In *Proceedings of the 2025 Conference on Creativity and Cognition*. 902–915.
- [19] Rudrajit Choudhuri, Bianca Trinkenreich, Rahul Pandita, Eirini Kalliamvakou, Igor Steinmacher, Marco Gerosa, Christopher Sanchez, and Anita Sarma. 2025. What Guides Our Choices? Modeling Developers' Trust and Behavioral Intentions Towards Genai. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE, 1691–1703.
- [20] Eric Corbett and Christopher Le Dantec. 2021. Designing civic technology with trust. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–17.
- [21] Eric Corbett and Christopher A Le Dantec. 2018. Going the distance: Trust work for citizen participation. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–13.
- [22] Ziyan Cui, Ning Li, and Huaikang Zhou. 2025. A large-scale replication of scenario-based experiments in psychology and management using large language models. *Nature Computational Science* (2025), 1–8.
- [23] Wesley Hanwen Deng, Wang Claire, Howard Ziyu Han, Jason I Hong, Kenneth Holstein, and Motahhare Eslami. 2025. WeAudit: Scaffolding User Auditors and AI Practitioners in Auditing Generative AI. *arXiv preprint arXiv:2501.01397* (2025).
- [24] Mateusz Dolata, Norbert Lange, and Gerhard Schwabe. 2024. Development in Times of Hype: How Freelancers Explore Generative AI? *2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE)* (2024), 2257–2269. <https://api.semanticscholar.org/CorpusId:266933328>

- [25] Yilun Du, Xueguang Ma, Shuran Song, Joshua B. Tenenbaum, and Antonio Torralba. 2024. Improving Factuality and Reasoning in Multi-agent Debate. In *Forty-first International Conference on Machine Learning (ICML)*.
- [26] 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* (2023).
- [27] Jan Gulliksen, Bengt Göransson, Inger Boivie, Stefan Blomkvist, Jenny Persson, and Åsa Cajander. 2003. Key principles for user-centred systems design. *Behaviour and Information Technology* 22, 6 (2003), 397–409.
- [28] Margaret Hagan. 2021. Prototyping for policy. In *Legal Design*. Edward Elgar Publishing, 9–31.
- [29] Howard Ziyu Han, Franklin Mingzhe Li, Alesandra Baca Vazquez, Daragh Byrne, Nikolas Martelaro, and Sarah E Fox. 2024. Co-design accessible public robots: Insights from people with mobility disability, robotic practitioners and their collaborations. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–20.
- [30] Björn Hartmann, Scott R Klemmer, Michael Bernstein, Leith Abdulla, Brandon Burr, Avi Robinson-Mosher, and Jennifer Gee. 2006. Reflective physical prototyping through integrated design, test, and analysis. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*. 299–308.
- [31] Peter Hase and Mohit Bansal. 2020. Evaluating Explainable AI: Which Algorithmic Explanations Help Users Predict Model Behavior?. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 5540–5552.
- [32] Patsy Healey, David E Booher, Jacob Torfing, Eva Sørensen, Mee Kam Ng, Pedro Peterson, and Louis Albrechts. 2008. Civic engagement, spatial planning and democracy as a way of life civic engagement and the quality of urban places enhancing effective and democratic governance through empowered participation: some critical reflections one humble journey towards planning for a more sustainable Hong Kong: a need to institutionalise civic engagement civic engagement and urban reform in Brazil setting the scene. *Planning Theory & Practice* 9, 3 (2008), 379–414.
- [33] Dirk Helbing, Hendrik Ammoser, and Christian Kühnert. 2006. Disasters as extreme events and the importance of network interactions for disaster response management. In *Extreme events in nature and society*. Springer, 319–348.
- [34] Dirk Helbing, Lubos Buzna, Anders Johansson, and Torsten Werner. 2005. Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions. *Transportation science* 39, 1 (2005), 1–24.
- [35] Dirk Helbing, Illes J Farkas, Peter Molnar, and Tamás Vicsek. 2002. Simulation of pedestrian crowds in normal and evacuation situations. *Pedestrian and evacuation dynamics* 21, 2 (2002), 21–58.
- [36] Dirk Helbing, Sachit Mahajan, Dino Carpentras, Monica Menendez, Evangelos Pournaras, Stefan Thurner, Trivik Verma, Elsa Arcaute, Michael Batty, and Luis MA Bettencourt. 2024. Co-creating the future: participatory cities and digital governance. *Philosophical Transactions A* 382, 2285 (2024), 20240113.
- [37] Kenneth Holstein, Bruce M. McLaren, and Vincent Aleven. 2019. Co-Designing a Real-Time Classroom Orchestration Tool to Support Teacher-AI Complementarity. *J. Learn. Anal.* 6 (2019). <https://api.semanticscholar.org/CorpusID:201128632>
- [38] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé III, Miro Dudík, and Hanna Wallach. 2019. Improving fairness in machine learning systems: What do industry practitioners need?. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–16.
- [39] John J Horton. 2023. *Large language models as simulated economic agents: What can we learn from homo silicus?* Technical Report. National Bureau of Economic Research.
- [40] Abe Bohan Hou, Hongru Du, Yichen Wang, Jingyu Zhang, Zixiao Wang, Paul Pu Liang, Daniel Khashabi, Lauren Gardner, and Tianxing He. 2025. Can A Society of Generative Agents Simulate Human Behavior and Inform Public Health Policy? A Case Study on Vaccine Hesitancy. *arXiv preprint arXiv:2503.09639* (2025).
- [41] V Daniel Hunt. 1996. *Process mapping: how to reengineer your business processes*. John Wiley & Sons.
- [42] Hilary Hutchinson, Wendy Mackay, Bo Westerlund, Benjamin B Bederson, Allison Druin, Catherine Plaisant, Michel Beaudouin-Lafon, Stéphane Conversy, Helen Evans, Heiko Hansen, et al. 2003. Technology probes: inspiring design for and with families. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 17–24.
- [43] Hilary Hutchinson, Wendy Mackay, Bo Westerlund, Benjamin B. Bederson, Allison Druin, Catherine Plaisant, Michel Beaudouin-Lafon, Stéphane Conversy, Helen Evans, Heiko Hansen, Nicolas Roussel, and Björn Eiderbäck. 2003. Technology probes: inspiring design for and with families. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA) (CHI '03). Association for Computing Machinery, New York, NY, USA, 17–24. doi:10.1145/642611.642616
- [44] Angel Hsing-Chi Hwang, Michael S Bernstein, S Shyam Sundar, Renwen Zhang, Manoel Horta Ribeiro, Yingdan Lu, Serina Chang, Tongshuang Wu, Aimee Yang, Dmitri Williams, et al. 2025. Human Subjects Research in the Age of Generative AI: Opportunities and Challenges of Applying LLM-Simulated Data to HCI Studies. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. 1–7.
- [45] Hyoungwook Jin, Minju Yoo, Jeongeon Park, Yokyung Lee, Xu Wang, and Juho Kim. 2025. TeachTune: Reviewing Pedagogical Agents Against Diverse Student Profiles with Simulated Students. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 1073, 28 pages. doi:10.1145/3706598.3714054
- [46] Shivani Kapania, William Agnew, Motahare Eslami, Hoda Heidari, and Sarah E Fox. 2025. Simulacrum of Stories: Examining Large Language Models as Qualitative Research Participants. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–17.
- [47] Seth Karten, Wenzhe Li, Zihan Ding, Samuel Kleiner, Yu Bai, and Chi Jin. 2025. LLM Economist: Large Population Models and Mechanism Design in Multi-Agent Generative Simulacra. *arXiv preprint arXiv:2507.15815* (2025).

- [48] Anna Kawakami, Amanda Coston, Haiyi Zhu, Hoda Heidari, and Kenneth Holstein. 2024. The Situate AI Guidebook: Co-Designing a Toolkit to Support Multi-Stakeholder, Early-stage Deliberations Around Public Sector AI Proposals. *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (2024). <https://api.semanticscholar.org/CorpusID:268063894>
- [49] Tzu-Sheng Kuo, Quan Ze Chen, Amy X Zhang, Jane Hsieh, Haiyi Zhu, and Kenneth Holstein. 2025. PolicyCraft: Supporting Collaborative and Participatory Policy Design through Case-Grounded Deliberation. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–24.
- [50] Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbulin, and Bernard Ghanem. 2023. CAMEL: Communicative Agents for “Mind” Exploration of Large Language Model Society. In *Advances in Neural Information Processing Systems*, A Oh, T Naumann, A Globerson, K Saenko, M Hardt, and S Levine (Eds.), Vol. 36. 51991–52008.
- [51] Junkai Li, Yung-hwei Lai, Weitao Li, Jingyi Ren, Meng Zhang, Xinhui Kang, Siyu Wang, Peng Li, Ya-Qin Zhang, Weizhi Ma, et al. 2024. Agent hospital: A simulacrum of hospital with evolvable medical agents. *arXiv preprint arXiv:2405.02957* (2024).
- [52] Xinyi Li, Yu Xu, Yongfeng Zhang, and Edward C Malthouse. 2024. Large language model-driven multi-agent simulation for news diffusion under different network structures. *arXiv preprint arXiv:2410.13909* (2024).
- [53] Yuxuan Li, Hirokazu Shirado, and Sauvik Das. 2025. Actions Speak Louder than Words: Agent Decisions Reveal Implicit Biases in Language Models. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency (FAccT ’25)*. Association for Computing Machinery, New York, NY, USA, 3303–3325. doi:10.1145/3715275.3732212
- [54] Charles Lindblom. 2018. The science of “muddling through”. In *Classic readings in urban planning*. Routledge, 31–40.
- [55] Joseph Lindley, Haider Ali Akmal, Franziska Pilling, and Paul Coulton. 2020. Researching AI legibility through design. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.
- [56] Genglin Liu, Vivian Le, Salman Rahman, Elisa Kreiss, Marzyeh Ghassemi, and Saadia Gabriel. 2025. Mosaic: Modeling social ai for content dissemination and regulation in multi-agent simulations. *arXiv preprint arXiv:2504.07830* (2025).
- [57] Jia'an Liu, Chu Chu, Yilin Zhao, Goshi Aoki, and Zhiqing Xiao. 2025. Agentic AI for Sustainable Development: Leveraging Large Language Model-Enhanced Agent-Based Modeling for Complex Policy Strategies. *Emerging Media* (2025), 27523543251365678.
- [58] Qun Ma, Xiao Xue, Deyu Zhou, Xiangning Yu, Donghua Liu, Xuwen Zhang, Zihan Zhao, Yifan Shen, Peilin Ji, Juanjuan Li, et al. 2024. Computational experiments meet large language model based agents: A survey and perspective. *arXiv preprint arXiv:2402.00262* (2024).
- [59] Dor Ma’ayan, Wode Ni, Katherine Ye, Chinmay Kulkarni, and Joshua Sunshine. 2020. How domain experts create conceptual diagrams and implications for tool design. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–14.
- [60] Robert Newcombe. 2003. From client to project stakeholders: a stakeholder mapping approach. *Construction management and economics* 21, 8 (2003), 841–848.
- [61] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. 2019. Dissecting racial bias in an algorithm used to manage the health of populations. *Science* 366, 6464 (2019), 447–453.
- [62] Jennifer O’Brien, Ellie Fossey, and Victoria J Palmer. 2021. A scoping review of the use of co-design methods with culturally and linguistically diverse communities to improve or adapt mental health services. *Health & Social Care in the Community* 29, 1 (2021), 1–17.
- [63] Leysia Palen and Sophia B Liu. 2007. Citizen communications in crisis: anticipating a future of ICT-supported public participation. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 727–736.
- [64] 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*. 1–22.
- [65] Joon Sung Park, Lindsay Popowski, Carrie Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2022. Social simulacra: Creating populated prototypes for social computing systems. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–18.
- [66] Joon Sung Park, Carolyn Q Zou, Aaron Shaw, Benjamin Mako Hill, Carrie Cai, Meredith Ringel Morris, Robb Willer, Percy Liang, and Michael S Bernstein. 2024. Generative agent simulations of 1,000 people. *arXiv preprint arXiv:2411.10109* (2024).
- [67] Jinghua Piao, Yuwei Yan, Jun Zhang, Nian Li, Junbo Yan, Xiaochong Lan, Zhihong Lu, Zhiheng Zheng, Jing Yi Wang, Di Zhou, et al. 2025. AgentSociety: Large-Scale Simulation of LLM-Driven Generative Agents Advances Understanding of Human Behaviors and Society. *arXiv preprint arXiv:2502.08691* (2025).
- [68] Elizabeth B-N Sanders and Pieter Jan Stappers. 2008. Co-creation and the new landscapes of design. *Co-design* 4, 1 (2008), 5–18.
- [69] Devansh Saxena, Seh Young Moon, Dahlia Shehata, and Shion Guha. 2022. Unpacking invisible work practices, constraints, and latent power relationships in child welfare through casenote analysis. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–22.
- [70] Douglas Schuler and Aki Namioka. 1993. *Participatory design: Principles and practices*. CRC press.
- [71] James Shore and Shane Warden. 2021. *The art of agile development*. " O'Reilly Media, Inc".
- [72] Peter Slattery, Alexander K Saeri, and Peter Bragge. 2020. Research co-design in health: a rapid overview of reviews. *Health research policy and systems* 18, 1 (2020), 17.
- [73] Kate Starbird and Leysia Palen. 2011. "Voluntweeters" self-organizing by digital volunteers in times of crisis. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 1071–1080.
- [74] Ian Steenstra, Farnaz Nouraei, and Timothy Bickmore. 2025. Scaffolding Empathy: Training Counselors with Simulated Patients and Utterance-level Performance Visualizations. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI ’25)*. Association for Computing

- Machinery, New York, NY, USA, Article 593, 22 pages. doi:10.1145/3706598.3714014
- [75] Anselm L Strauss. 1987. *Qualitative analysis for social scientists*. Cambridge university press.
- [76] Theodore Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas Griffiths. 2023. Cognitive architectures for language agents. *Transactions on Machine Learning Research* (2023).
- [77] Andrea K. Thomer, Michael B. Twidale, Jinlong Guo, and Matthew J. Yoder. 2016. Co-designing Scientific Software: Hackathons for Participatory Interface Design. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (San Jose, California, USA) (*CHI EA '16*). Association for Computing Machinery, New York, NY, USA, 3219–3226. doi:10.1145/2851581.2892549
- [78] Sarah Vieweg, Amanda L Hughes, Kate Starbird, and Leysia Palen. 2010. Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 1079–1088.
- [79] Vasillis Vlachokyriakos, Clara Crivellaro, Christopher A Le Dantec, Eric Gordon, Pete Wright, and Patrick Olivier. 2016. Digital civics: Citizen empowerment with and through technology. In *Proceedings of the 2016 CHI conference extended abstracts on human factors in computing systems*. 1096–1099.
- [80] Karel Vredenburg, Ji-Ye Mao, Paul W Smith, and Tom Carey. 2002. A survey of user-centered design practice. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 471–478.
- [81] Dakuo Wang, Josh Andres, Justin D Weisz, Erick Oduor, and Casey Dugan. 2021. Autods: Towards human-centered automation of data science. In *Proceedings of the 2021 CHI conference on human factors in computing systems*. 1–12.
- [82] Katharina Weitz, Ruben Schlagowski, Elisabeth André, Maris Männiste, and Ceenu George. 2024. Explaining It Your Way - Findings from a Co-Creative Design Workshop on Designing XAI Applications with AI End-Users from the Public Sector. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (*CHI '24*). Association for Computing Machinery, New York, NY, USA, Article 745, 14 pages. doi:10.1145/3613904.3642563
- [83] Ross Williams, Niyousha Hosseinichimeh, Aritra Majumdar, and Navid Ghaffarzadegan. 2023. Epidemic modeling with generative agents. *arXiv preprint arXiv:2307.04986* (2023).
- [84] Austin P Wright, Omar Shaikh, Haekyu Park, Will Epperson, Muhammed Ahmed, Stephane Pinel, Duen Horng Chau, and Diyi Yang. 2021. RECAST: Enabling user recourse and interpretability of toxicity detection models with interactive visualization. *Proceedings of the ACM on human-computer interaction* 5, CSCW1 (2021), 1–26.
- [85] Chengxing Xie, Canyu Chen, Feiran Jia, Ziyu Ye, Shiyang Lai, Kai Shu, Jindong Gu, Adel Bibi, Ziniu Hu, David Jurgens, et al. 2024. Can Large Language Model Agents Simulate Human Trust Behavior?. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- [86] Fengli Xu, Jun Zhang, Chen Gao, Jie Feng, and Yong Li. 2023. Urban generative intelligence (ugi): A foundational platform for agents in embodied city environment. *arXiv preprint arXiv:2312.11813* (2023).
- [87] Qian Yang, Richmond Y Wong, Steven Jackson, Sabine Junginger, Margaret D Hagan, Thomas Gilbert, and John Zimmerman. 2024. The future of HCI-policy collaboration. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–15.
- [88] Zheyuan Zhang, Daniel Zhang-Li, Jifan Yu, Linlu Gong, Jinchang Zhou, Zhanxin Hao, Jianxiao Jiang, Jie Cao, Huiqin Liu, Zhiyuan Liu, et al. 2024. Simulating classroom education with llm-empowered agents. *arXiv preprint arXiv:2406.19226* (2024).
- [89] Xuhui Zhou, Zhe Su, Tiwalayo Eisape, Hyunwoo Kim, and Maarten Sap. 2024. Is this the real life? is this just fantasy? the misleading success of simulating social interactions with llms. *arXiv preprint arXiv:2403.05020* (2024).
- [90] John Zimmerman, Jodi Forlizzi, and Shelley Evenson. 2007. Research through design as a method for interaction design research in HCI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) (*CHI '07*). Association for Computing Machinery, New York, NY, USA, 493–502. doi:10.1145/1240624.1240704

## A Appendix

#### A.1 Co-creating Stakeholder Maps and Process Maps

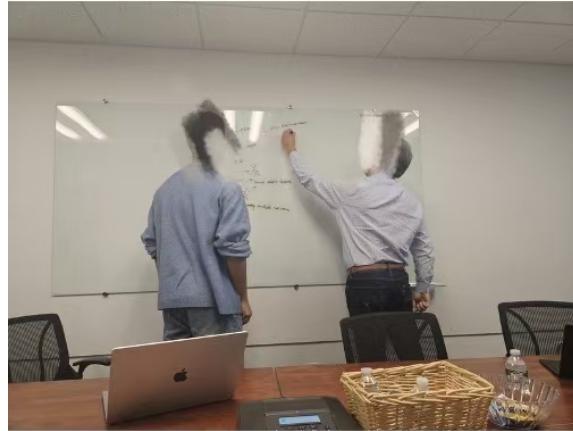


Fig. A1. Developers and policymakers *P1* collaboratively co-creating a stakeholder map and process map during a policy meeting. The session supported knowledge elicitation of roles, responsibilities, and workflows in emergency preparedness.

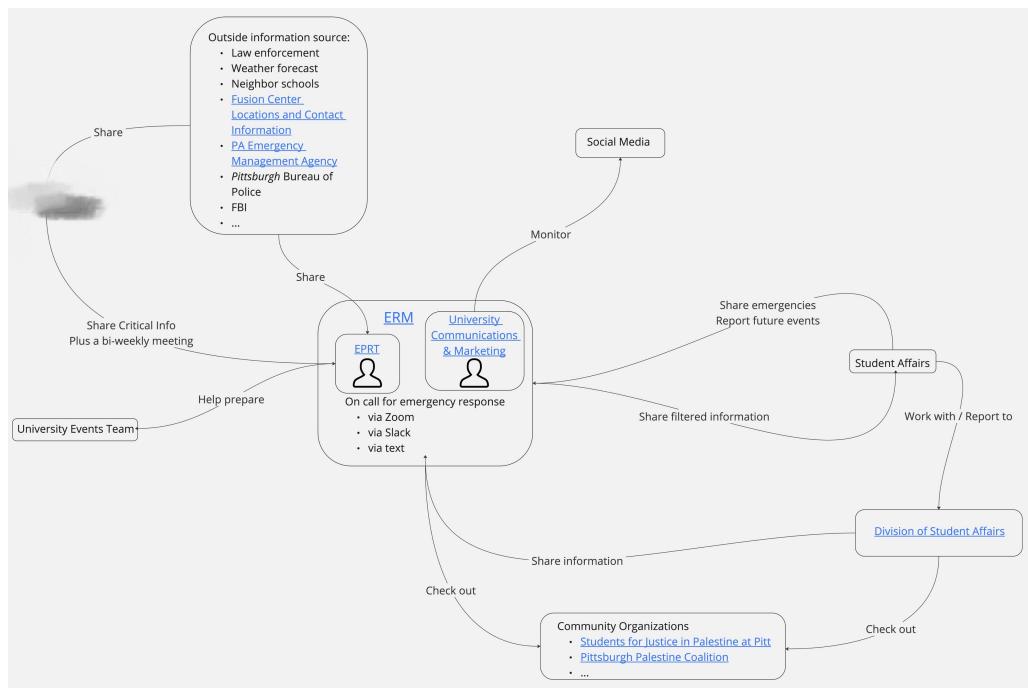


Fig. A2. Stakeholder map of emergency preparedness and response at the university, co-created with policymakers to identify actors, roles, and relationships shaping decision-making and information flows during crises.

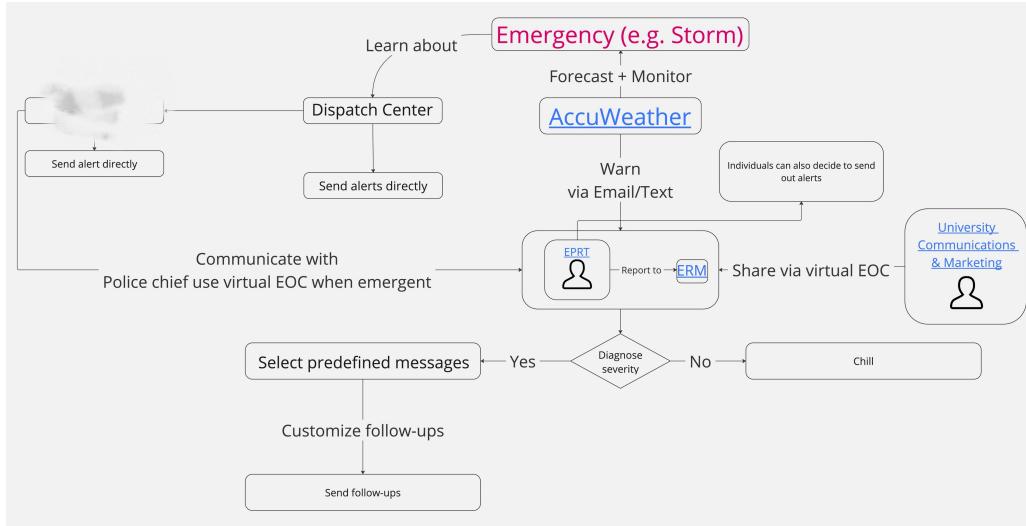


Fig. A3. Process diagram of emergency preparedness and response at the university, documenting sequential phases from preparedness planning to active response and recovery, and highlighting interdependencies among roles.

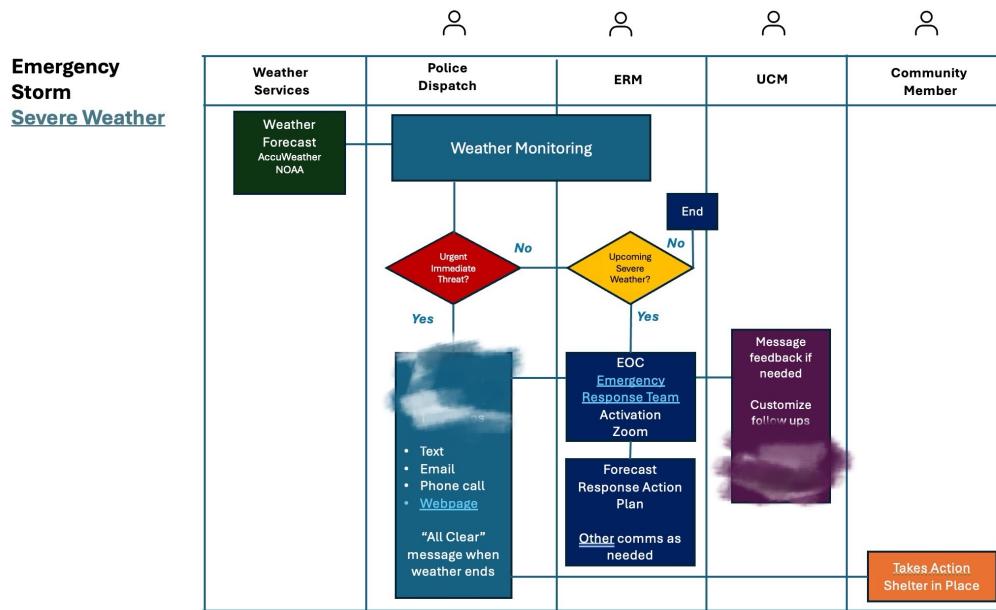


Fig. A4. Process diagram of emergency preparedness and response tailored for severe weather scenarios, illustrating protocols for monitoring, issuing warnings, coordinating shelter, and resuming normal operations.

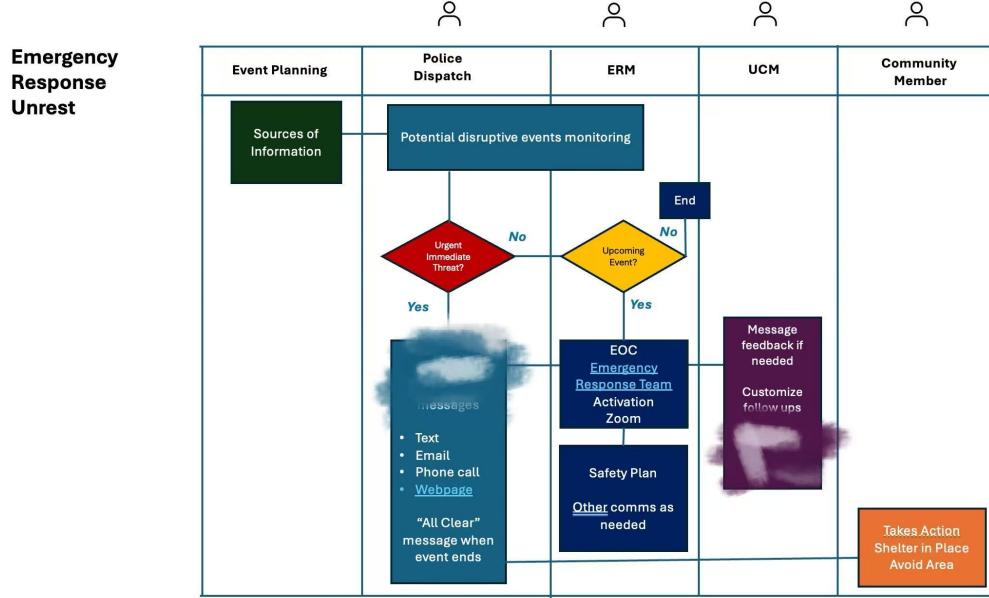


Fig. A5. Process diagram of emergency preparedness and response tailored for unrest or bomb threat scenarios, showing decision pathways for lockdowns, communication flows, and coordination with law enforcement.

## A.2 System Description: Iteration 1

**A.2.1 Misinterpretation Simulation System.** The simulation system implements a multi-stage pipeline designed to evaluate how university announcements may be misinterpreted by students with diverse cognitive and behavioral characteristics. The system operates through four primary phases: agent generation, message creation, interpretation simulation, and misinterpretation assessment.

**Agent generation and characterization.** The system begins by generating a diverse population of simulated students, each represented as an agent with distinct personal characteristics. Each agent is assigned a randomized name and a detailed persona that encompasses various psychological and behavioral traits. The persona specifications explicitly include the individual's propensity for misinterpreting information, ranging from highly analytical students who carefully parse communications to individuals who may jump to conclusions or misread contextual cues. Additional persona elements may include academic background, stress levels, prior experiences with institutional communications, attention to detail, and emotional reactivity patterns.

**Message corpus development.** In parallel to agent creation, the system generates a corpus of representative university announcement messages. These messages are designed to reflect typical institutional communications that students might receive, spanning various topics such as policy changes, event notifications, academic deadlines, or administrative updates.

**Interpretation and reaction simulation.** The core simulation phase involves exposing each generated agent to each message in the corpus. For every agent-message pair, the system simulates the student's cognitive processing by generating two outputs: an interpretation of the message based on the agent's persona, and a behavioral reaction stemming from that interpretation. The interpretation represents the agent's internal understanding of the message

content, filtered through their individual characteristics and biases. The reaction component captures how the student would likely respond to the message given their specific interpretation and personality traits.

**Misinterpretation scoring and assessment.** Following the interpretation simulation, the system employs an independent assessment mechanism to evaluate the degree of misinterpretation present in each agent's response. This assessment compares the agent's interpretation and reaction against the original message content to assign a numerical misinterpretation score. The scoring system operates on a scale from zero to one hundred, where zero indicates perfect comprehension and one hundred represents complete misunderstanding of the intended message.

**Extreme response generation.** For cases where the misinterpretation score exceeds a threshold value, indicating significant misunderstanding, the system generates an additional extreme reaction scenario. This component simulates how the student might respond if their misinterpretation led to heightened emotional or behavioral responses, providing insight into potential worst-case scenarios for communication failures.

---

**Algorithm 1** Pseudo Script: Misinterpretation Simulation System

---

**Require:** Number of agents  $N$ , Number of messages  $M$   
**Ensure:** Misinterpretation analysis results for all agent-message pairs

```

1: Phase 1: Agent and Message Generation
2:  $A \leftarrow \{\}$                                      ▷ Initialize agent set
3: for  $i = 1 \rightarrow N$  do
4:    $name_i \leftarrow \text{GenerateRandomName}()$ 
5:    $persona_i \leftarrow \text{GeneratePersona(misinterpretation\_tendency)}$ 
6:    $A \leftarrow A \cup \{(name_i, persona_i)\}$ 
7:  $Messages \leftarrow \text{GenerateMessageCorpus}(M)$            ▷ Generate university announcements
8: Phase 2: Interpretation Simulation
9:  $Results \leftarrow \{\}$                                      ▷ Initialize results storage
10: for all  $message_j \in Messages$  do
11:   for all  $agent_i \in A$  do
12:      $interpretation_{i,j} \leftarrow \text{SimulateInterpretation}(agent_i, message_j)$ 
13:      $reaction_{i,j} \leftarrow \text{SimulateReaction}(agent_i, message_j, interpretation_{i,j})$ 
14:   Phase 3: Misinterpretation Assessment
15:    $score_{i,j} \leftarrow \text{AssessMisinterpretation}(message_j, interpretation_{i,j}, reaction_{i,j})$ 
16:   Phase 4: Extreme Response Generation
17:   if  $score_{i,j} > \text{threshold}$  then
18:      $extreme_{i,j} \leftarrow \text{GenerateExtremeReaction}(agent_i, message_j, interpretation_{i,j})$ 
19:   else
20:      $extreme_{i,j} \leftarrow \emptyset$ 
21:    $Results[j] \leftarrow Results[j] \cup \{(agent_i, interpretation_{i,j}, reaction_{i,j}, score_{i,j}, extreme_{i,j})\}$ 
return  $Results$                                      ▷ Complete misinterpretation analysis dataset

```

---

**A.2.2 Propagation Simulation System.** The LLM agent simulation framework implements a multi-agent system designed to model information propagation and collective decision-making behaviors in response to security-related scenarios. The system consists of three core components: autonomous agents with individual personas, information diffusion mechanisms, and environmental orchestration.

**Agent architecture and decision-making.** Each agent possesses a unique identity with a name and detailed persona that influences their decision-making patterns. Agents operate through a structured three-phase decision

pipeline when processing information. In the decision phase, agents evaluate all received information using their persona and accumulated knowledge to determine their reaction from three possible states: idle behavior, active information spreading, or evacuation from the simulation. When agents choose to spread information, they enter an action phase where they generate original content based on their understanding of the situation and their personal characteristics. Finally, agents undergo an update phase where they incorporate feedback about simulation outcomes into their knowledge base, enabling learning and adaptation for future scenarios.

The agent decision-making process utilizes large language models with structured prompts tailored to each agent's persona. Agents maintain separate conversation histories for each decision type, ensuring contextual consistency while preventing interference between different cognitive processes. The reward mechanism allows agents to accumulate insights from environmental feedback, creating memory effects where past experiences influence future behavior patterns. Agents can transition between active and inactive states, with evacuation representing a permanent withdrawal from the simulation.

**Information diffusion mechanisms.** The system implements two distinct diffusion strategies to study the effects of content moderation on information propagation. The moderated diffusion approach intercepts all agent-generated content and applies standardized warning labels indicating potential misinformation before distribution. Specifically, content is prefixed with explicit warnings stating "WARNING: This piece of content might contain misinformation" followed by the original agent message. The unmoderated diffusion approach transmits agent-generated content without any modification, serving as a control condition.

Both diffusion mechanisms employ identical probabilistic distribution patterns. When an agent selects the spreading action, the diffusion system randomly samples 70% of all agents in the simulation as recipients for the generated content. This sampling approach creates realistic information propagation patterns where content does not reach the entire population simultaneously. The 70% coverage rate ensures substantial information spread while maintaining realistic constraints on individual agent reach within social networks.

**Environmental coordination and scenario management.** The environmental system orchestrates complex multi-round simulations through structured scenario scripts that define information injection patterns, timing, and feedback mechanisms. These scripts specify which agents receive initial information, the source attribution for credibility modeling (such as news organizations or individual sources), and the specific content payloads representing rumors, official announcements, or other information types.

Simulation execution proceeds through discrete rounds where the environment first distributes scripted information to designated agent subsets. Active agents then process their inputs and make decisions according to their decision pipeline. Any agents choosing to spread information generate content that enters the diffusion mechanism, creating secondary information waves that become inputs for subsequent rounds. This process continues until all scripted rounds complete, generating cascading information effects throughout the agent population.

The system implements section-based feedback where agents receive environmental outcomes based on their final states at the end of each scenario section. Agents learn whether their decisions were appropriate given the actual nature of the security threat, enabling behavioral adaptation across multiple scenario sections.

**Information propagation dynamics and network effects.** Information flow within the system emerges organically from agent decisions rather than following predefined network topologies. The system creates dynamic communication networks where information pathways form based on which agents choose to spread content and which agents are randomly selected as recipients. This approach models realistic social media environments where information sharing creates temporary communication channels between individuals.

The propagation mechanism generates compound effects where initial information triggers multiple rounds of agent responses, decisions, and subsequent information generation. Agent-generated content influences recipient decision-making, potentially creating viral information cascades where rumors or alerts spread exponentially through the population. The random sampling approach ensures that different agents may receive different combinations of information, leading to diverse response patterns and realistic heterogeneity in population-level outcomes.

---

**Algorithm 2** Pseudo Script: Propagation Simulation System
 

---

```

Require:  $A = \{a_1, a_2, \dots, a_n\}$                                 ▷ Set of agents with personas
Require:  $S$                                               ▷ Scenario script with sections and rounds
Require:  $D$                                               ▷ Diffusion mechanism (moderated or unmoderated)

1: Initialize all agents  $a_i$  with personas, decision histories, and reward sets
2: Set all agent states to ACTIVE
3: for each section  $s$  in  $S$  do
4:    $propagated\_info \leftarrow \emptyset$ 
5:   for each round  $r$  in  $s$  do
6:      $current\_round\_actions \leftarrow \emptyset$ 
7:      $round\_inputs \leftarrow \text{Combine}(r, propagated\_info)$           ▷ Merge scripted and propagated info
8:     for each agent  $a_i \in A$  where  $\text{state}(a_i) = \text{ACTIVE}$  and  $a_i$  has inputs do
9:        $inputs_i \leftarrow round\_inputs[a_i]$ 
10:       $context \leftarrow \text{BuildContext}(a_i.\text{persona}, a_i.\text{rewards}, inputs_i)$ 
11:       $decision_i \leftarrow \text{LLMCall}(a_i.\text{decision\_history}, context)$ 
12:      if  $decision_i = \text{SPREAD}$  then
13:         $content_i \leftarrow \text{LLMCall}(a_i.\text{action\_history}, context)$ 
14:        Add  $\{\text{role} : a_i.\text{name}, \text{content} : content_i\}$  to  $current\_round\_actions$ 
15:      else if  $decision_i = \text{EVACUATE}$  then
16:         $\text{state}(a_i) \leftarrow \text{INACTIVE}$ 
17:       $propagated\_info \leftarrow \emptyset$ 
18:      for each action  $act$  in  $current\_round\_actions$  do
19:         $recipients \leftarrow \text{RandomSample}(A, 0.7 \times |A|)$           ▷ 70% coverage
20:        if  $D$  is moderated then
21:           $content \leftarrow \text{"WARNING: misinformation. "} + act.\text{content}$ 
22:        else
23:           $content \leftarrow act.\text{content}$ 
24:        Add  $\{recipients : recipients, source : act.\text{role}, content : content\}$  to  $propagated\_info$ 
25:      if  $s$  is not the last section then
26:        for each agent  $a_i \in A$  do
27:           $result_i \leftarrow \text{GetSectionResult}(s, state(a_i))$ 
28:           $knowledge \leftarrow \text{LLMCall}(a_i.\text{update\_history}, \text{BuildContext}(result_i))$ 
29:           $a_i.\text{rewards} \leftarrow knowledge$ 
30:        Clear agent histories and reset  $\text{state}(a_i) \leftarrow \text{ACTIVE}$ 

```

---

### A.3 System Description: Final Iteration

**A.3.1 Agent Population Generation.** The simulation employs a **two-stage persona generation process** to create realistic and diverse agent populations representing commencement attendees. The first stage generates individual student personas through parallel API calls to language models, while the second stage creates accompanying family and friend personas for students requiring social connections.

Student persona generation operates across ten academic disciplines with predetermined enrollment distributions, totaling 2,928 base student personas. Each generation request includes the academic major as context to produce discipline-appropriate background profiles. The system generates an additional 44 students with accessibility requirements, distributed proportionally across all academic programs.

The persona generation algorithm assigns unique demographic characteristics including names, detailed background descriptions encompassing academic interests and personal circumstances, accessibility requirements, and group affiliation patterns. Students with accessibility needs receive explicit mobility considerations that influence their movement parameters during simulation execution.

The second generation stage creates social network structures through a stratified assignment process. The system randomly partitions the 2,928 students into three categories: 2,000 students who attend with family members or friends (requiring additional persona generation), 800 students who form friend groups with other students, and 128 students who attend individually. For the family/friend category, the system generates between 1-8 additional personas per student using targeted prompts that specify relationship types and demographic coherence with the primary student.

Friend group formation employs stochastic clustering with group sizes ranging from 3-10 members, randomly sampling from the designated student pool until all members receive group assignments. Each generated group establishes bidirectional membership lists that define communication channels and decision-making units during simulation execution.

The complete population generation process produces approximately 13,000 total agents with explicit social network topologies, realistic demographic distributions, and heterogeneous accessibility requirements that directly influence simulation behavior patterns.

**A.3.2 Agent Architecture and Initialization.** The simulation instantiates individual agents representing commencement attendees, each configured with distinct behavioral parameters and social network affiliations. Each agent maintains a comprehensive profile including demographic information, academic major, accessibility requirements, and explicit group membership identifiers that define their social connections within the simulation.

The system categorizes agents into four distinct behavioral classes that determine their decision-making protocols: students with family and friends outside the stadium, students with friends inside the venue, students attending alone, and family members or friends of graduates. These classifications directly influence the agent's response patterns and group interaction capabilities.

Agent initialization occurs at predetermined coordinates within a  $2400 \times 1200$  pixel discrete coordinate system representing the stadium layout. The coordinate space encompasses geometrically defined regions including eight numbered seating sections arranged in a  $2 \times 4$  grid, interconnecting pathways with specified widths, designated family and accessibility areas positioned along the stadium perimeter, a central rectangular stage obstacle, and four exit points located at coordinates (20,20), (2380,20), (20,1180), and (2380,600).

Each agent operates with a 20-pixel visibility radius for environmental detection and maintains internal state variables tracking their current position, movement target coordinates, decision history, and group chat message logs. The system assigns each agent an accessibility flag that modifies their movement parameters and a visibility radius parameter that determines their environmental awareness range.

**A.3.3 Decision-Making Process and Response Formats.** The simulation implements two distinct decision-making protocols differentiated by agent classification and social context. Non-student-alone agents participate in structured group discussion rounds using a multi-field response format that captures both individual decision state and social

**Algorithm 3** Commencement Emergency Response Simulation System (Part 1: Initialization & Population)

---

```

1: procedure INITIALIZESIMULATION
2:   canvas  $\leftarrow$  Create2DSpace(2400, 1200)
3:   stadium_features  $\leftarrow$  DefineStadiumLayout()
4:   coordinators  $\leftarrow$  PlaceCoordinators(50, canvas)
5:   agents  $\leftarrow$  GenerateAgentPopulation()
6:   round  $\leftarrow$  0
7:   return agents, canvas, coordinators
8: procedure GENERATEAGENTPOPULATION
9:   students  $\leftarrow$   $\emptyset$ 
10:  for major in {Engineering(720), Business(240), ...} do
11:    personas  $\leftarrow$  GeneratePersonasAsync(major)
12:    students  $\leftarrow$  students  $\cup$  personas
13:  population  $\leftarrow$  CreateSocialNetworks(students)
14:  return population
15: procedure CREATSOCIALNETWORKS(students)
16:   others_list, friends_list, alone_list  $\leftarrow$  RandomPartition(students, [2000, 800, 128])
17:   for student in others_list do
18:     family_count  $\leftarrow$  Random(1, 8)
19:     family  $\leftarrow$  GenerateFamilyPersonas(student, family_count)
20:     SetGroupMembership(student  $\cup$  family)
21:   for remaining in friends_list do
22:     group_size  $\leftarrow$  Random(3, 10)
23:     group  $\leftarrow$  Sample(remaining, group_size)
24:     SetGroupMembership(group)
25:   return others_list  $\cup$  friends_list  $\cup$  alone_list

```

---

communication. These agents generate responses containing a boolean decision indicator, an optional destination selection from a predefined enumeration of twelve possible locations, and a natural language message for intra-group communication.

The destination enumeration includes four exit locations (Exit 1 through Exit 4), five stadium region descriptions (North/South/West/East track areas, South bleachers area), and three social gathering areas (West/East family and friends areas, West seating sections area). Each destination maps to specific coordinate generation algorithms that produce either fixed exit coordinates or randomized positions within defined regional boundaries.

Student-alone agents bypass group consultation mechanisms and utilize a simplified response protocol that produces only a destination selection without accompanying social messaging. This streamlined process eliminates group coordination overhead while maintaining equivalent environmental input processing.

Both decision protocols receive identical contextual information packages generated through environmental analysis algorithms. These packages contain structured descriptions of the agent's current stadium location, enumerated nearby physical features within the visibility radius, directional and distance information for all visible agents, coordinator proximity notifications with specific exit recommendations, and ranked distance calculations to all four exits with cardinal direction indicators.

*A.3.4 Environmental Context Generation.* The simulation constructs detailed environmental descriptions through systematic analysis of agent position relative to stadium features and other attendees. The context generation algorithm

first identifies the agent's current location by testing point-in-rectangle containment against all defined stadium features, producing location-specific descriptions that vary based on feature type.

For agents positioned within seating sections, the system calculates row and column positions using coordinate offset arithmetic and unit spacing parameters. Agents on pathways receive directional information about pathway endpoints and connecting features. The algorithm generates specialized descriptions for agents in family areas, accessibility zones, or open spaces between major features.

Proximity detection operates through distance calculations between agent positions and all stadium features, filtering results by the agent's visibility radius. The system computes cardinal directions using arctangent calculations and maps angular ranges to eight-direction compass bearings. Distance measurements undergo categorical classification into descriptive ranges: extremely close (less than 50 pixels), near (50-150 pixels), moderately far (150-400 pixels), far (400-800 pixels), and very far (greater than 800 pixels).

The system performs comprehensive exit ranking by calculating Euclidean distances from the agent's position to all four exit coordinates, sorting results by proximity, and generating formatted descriptions that include both distance categories and directional bearings. This exit information provides agents with consistent spatial orientation data for navigation decision-making.

---

**Algorithm 4** Commencement Emergency Response Simulation System (Part 2a: Main Loop Core)

---

```

1: procedure SIMULATIONMAINLOOP(agents, canvas, coordinators)
2:   while  $\exists \text{agent} \in \text{agents} : \text{agent.state} \neq \text{EXITED}$  do
3:     round  $\leftarrow$  round + 1
                                          $\triangleright$  Phase 1: State Transition Analysis
4:     resumed_agents  $\leftarrow$  CheckGroupArrivals(agents)
5:     influenced_groups  $\leftarrow$  CheckCoordinatorInfluence(agents, coordinators)
6:     ResetInfluencedGroups(influenced_groups)
                                          $\triangleright$  Phase 2: Decision Processing
7:     discussing_agents  $\leftarrow \{a \in \text{agents} : a.\text{state} = \text{DISCUSSING}\}$ 
8:     decisions  $\leftarrow$  ProcessDecisionsConcurrently(discussing_agents)
9:     UpdateAgentStates(decisions)
                                          $\triangleright$  Phase 3: Movement Processing
10:    moving_agents  $\leftarrow \{a \in \text{agents} : a.\text{state} = \text{MOVING}\}$ 
11:    density_map  $\leftarrow$  CalculateDensity(moving_agents)
12:    ExecuteMovement(moving_agents, density_map)
13:    CheckDestinationArrivals(moving_agents)
                                          $\triangleright$  Phase 4: Data Logging
14:    LogRoundData(round, agents, decisions)

```

---

**A.3.5 Movement Mechanics and Pathfinding.** Agent movement operates through a coordinate-based pathfinding system that translates abstract destination selections into specific target coordinates and executes movement through obstacle-aware navigation algorithms. The destination resolution process maps each of the twelve possible destination types to coordinate generation functions that produce either deterministic exit positions or stochastic coordinates within defined regional boundaries.

Regional destinations employ bounded random coordinate generation with feature-aware constraints. For example, "North side of the stadium, track area" generates coordinates within the northern boundary rectangle while explicitly avoiding intersection with the central stage obstacle through point-in-rectangle collision detection.

The movement calculation system processes agent motion through a multi-stage algorithm that accounts for movement constraints, obstacle avoidance, and environmental factors. Base movement speed varies according to agent characteristics: standard agents move at 24 pixels per round, accessibility-flagged agents move at 16 pixels per round, and all agents receive speed modifications based on environmental conditions.

Density-based speed adjustment operates through a graduated reduction system that counts nearby agents within a configurable radius and applies speed penalties. The algorithm maintains maximum speed when fewer than 4 agents occupy the proximity zone, linearly reduces speed as nearby agent count increases toward 30 agents, and enforces minimum speed thresholds to prevent complete movement cessation in crowded conditions.

Coordinator proximity provides speed enhancement, increasing movement rates to 32 pixels per round for standard agents and 16 pixels per round for accessibility agents when within 50 pixels of any coordinator entity. This mechanism simulates urgency responses to authority figure instructions.

The pathfinding algorithm implements vector-based movement with obstacle sliding behavior. The system calculates unit vectors toward target coordinates, detects intersection points with obstacle boundaries using line-segment-rectangle intersection calculations, and applies sliding movement along obstacle edges when direct paths are blocked. Special handling prevents agents from entering the stage area unless they begin their movement within that region.

*A.3.6 Coordinator Influence and Behavioral Modification.* The simulation deploys 50 coordinator entities at predetermined strategic positions throughout the stadium layout, each configured with specific exit recommendation parameters. Coordinator placement follows a spatial distribution pattern: northern coordinators recommend Exit 1, southern coordinators suggest Exit 3, with positioning designed to provide coverage across major circulation areas including pathway intersections, seating section perimeters, and family gathering zones.

The coordinator influence mechanism operates through proximity detection coupled with destination conflict analysis. When agents enter the 50-pixel influence radius around any coordinator, the system compares the agent's current destination against the coordinator's programmed exit recommendation. Destination mismatches trigger the influence protocol, which applies probabilistic behavioral modification based on a configurable reaction probability parameter.

Upon influence activation, the system implements group-level behavioral reset mechanisms. For student-alone agents, the influence directly affects the individual agent. For group-affiliated agents, the influence cascades to all members of the agent's social group, regardless of their individual proximity to coordinators. This collective influence model simulates realistic group decision-making dynamics where authority recommendations affect entire social units.

The behavioral reset process transitions all affected agents from their current movement or waiting states back to discussion state, clears their existing destination selections and movement targets, and injects the coordinator's exit recommendation into their next decision-making round as environmental context. The system tracks these influence events in the simulation log with detailed records of affected agents, original destinations, coordinator suggestions, and agent positions at the time of influence.

*A.3.7 State Management and Group Coordination.* Agent state management operates through a four-state finite state machine with explicit transition conditions and group synchronization mechanisms. The states comprise: discussing (participating in decision-making rounds), moving (navigating toward selected destinations), waiting (paused at intermediate destinations), and exited (reached final exit and removed from active simulation).

The group coordination system maintains data structures tracking destination selections and arrival status for each social group, identified by sorted member ID tuples. When agents select destinations, the system records both the

**Algorithm 5** Commencement Emergency Response Simulation System (Part 2b: Coordinator Influence & Decisions)

---

```

1: procedure CHECKCOORDINATORINFLUENCE(agents, coordinators)
2:   influenced_groups  $\leftarrow \emptyset$ 
3:   for agent in  $\{a \in \text{agents} : a.\text{state} = \text{MOVING}\}$  do
4:     for coord in coordinators do
5:       if Distance(agent.position, coord.position)  $\leq 50$  then
6:         if agent.destination  $\neq \text{coord.suggested\_exit}$  then
7:           if Random()  $< 0.5$  then
8:             influenced_groups  $\leftarrow \text{influenced\_groups} \cup \{\text{agent.group}\}$ 
9:   return influenced_groups

10: procedure PROCESSDECISIONSCONCURRENTLY(discussing_agents)
11:   tasks  $\leftarrow \emptyset$ 
12:   for agent in discussing_agents do
13:     context  $\leftarrow \text{GenerateEnvironmentalContext}(\text{agent})$ 
14:     if agent.type = student_alone then
15:       task  $\leftarrow \text{CreateDecisionTask}(\text{agent}, \text{context})$ 
16:     else
17:       group_messages  $\leftarrow \text{GetGroupChatHistory}(\text{agent.group})$ 
18:       task  $\leftarrow \text{CreateDiscussionTask}(\text{agent}, \text{context}, \text{group\_messages})$ 
19:     tasks  $\leftarrow \text{tasks} \cup \{\text{task}\}$ 
20:   responses  $\leftarrow \text{ExecuteTasksConcurrently}(\text{tasks}, \text{max\_concurrent}=2000)$ 
21:   return responses

22: procedure GENERATEENVIRONMENTALCONTEXT(agent)
23:   current_location  $\leftarrow \text{IdentifyStadiumFeature}(\text{agent.position})$ 
24:   nearby_features  $\leftarrow \text{FindFeaturesInRadius}(\text{agent.position}, 20)$ 
25:   nearby_agents  $\leftarrow \text{FindAgentsInRadius}(\text{agent.position}, 20)$ 
26:   exit_rankings  $\leftarrow \text{RankExitsByDistance}(\text{agent.position})$ 
27:   coordinator_info  $\leftarrow \text{CheckCoordinatorProximity}(\text{agent.position})$ 
28:   description  $\leftarrow \text{GenerateNaturalLanguageDescription}(\text{current\_location}, \text{nearby\_features}, \text{nearby\_agents}, \text{exit\_rankings}, \text{coordinator\_info})$ 
29:   return description

```

---

choice and the selecting agent in group-specific tracking tables. As agents reach their target coordinates (determined by Euclidean distance calculations with a 50-pixel tolerance threshold), the system updates arrival status records.

Automatic discussion resumption occurs when all group members who selected identical intermediate destinations successfully arrive at those locations. The coordination algorithm periodically scans all tracked groups, identifies destinations where the set of selecting agents equals the set of arrived agents, and transitions all relevant agents from waiting state back to discussing state. This mechanism ensures group cohesion during multi-stage navigation without requiring explicit coordination messages.

Special handling accommodates student-alone agents who bypass group coordination requirements. These agents automatically transition from waiting to discussing states without requiring group member synchronization, preventing indefinite waiting conditions for agents without social dependencies.

**A.3.8 Round-Based Execution and Concurrency Management.** The simulation operates through discrete **round-based execution cycles** that process agent decisions and movements in structured phases. Each round begins with state Manuscript submitted to ACM

transition analysis, identifying agents eligible for discussion resumption based on group arrival status and processing coordinator influence effects on currently moving agents.

The discussion processing phase handles all agents in discussing state through concurrent API request management. The system implements semaphore-based concurrency control with configurable limits (defaulting to 2000 concurrent requests) and rate limiting mechanisms to manage external language model API interactions. Discussion tasks execute asynchronously with timeout handling and retry logic for failed requests.

Response processing extracts structured decision data from language model outputs, updates agent internal states based on destination selections, and maintains group chat message histories for subsequent rounds. The system validates destination selections against the predefined enumeration and applies fuzzy string matching to handle response variations.

Movement processing operates on all agents in moving state simultaneously, calculating density-based speed adjustments through spatial proximity analysis and updating agent coordinates via pathfinding algorithms. The system performs destination arrival checking after position updates and transitions agents to appropriate successor states based on destination types (exits trigger exited state, intermediate locations trigger waiting state).

Data logging captures comprehensive round information including agent positions, state classifications, group messages, individual decisions, coordinator influence events, and API interaction statistics. The system maintains both in-memory state for active simulation processing and persistent JSON output for subsequent analysis, with intermediate result saving to prevent data loss during extended simulation runs.

The simulation continues until all agents reach exited state or a predefined maximum round limit is exceeded, providing natural termination conditions for both successful evacuation scenarios and extended simulation studies.

---

**Algorithm 6** Commencement Emergency Response Simulation System (Part 3a: Movement & Speed)
 

---

```

1: procedure EXECUTEMOVEMENT(moving_agents, density_map)
2:   for agent in moving_agents do
3:     nearby_count  $\leftarrow$  density_map[agent.id]
4:     speed  $\leftarrow$  CalculateAdjustedSpeed(agent, nearby_count)
5:     new_position  $\leftarrow$  PathfindToTarget(agent.position, agent.target, speed)
6:     agent.position  $\leftarrow$  new_position
7: procedure CALCULATEADJUSTEDSPEED(agent, nearby_count)
8:   if agent.accessibility = true then
9:     base_speed  $\leftarrow$  16, min_speed  $\leftarrow$  4.8
10:   else
11:     base_speed  $\leftarrow$  24, min_speed  $\leftarrow$  6.4
12:   if CoordinatorNearby(agent) then
13:     base_speed  $\leftarrow$  base_speed  $\times$  1.33
14:   if nearby_count  $\leq$  4 then
15:     speed_factor  $\leftarrow$  1.0
16:   else if nearby_count  $\geq$  30 then
17:     speed_factor  $\leftarrow$  0.0
18:   else
19:     speed_factor  $\leftarrow$   $1.0 - \frac{\text{nearby\_count} - 4}{30 - 4}$ 
20:   adjusted_speed  $\leftarrow$  max(min_speed, base_speed  $\times$  speed_factor)
21:   return adjusted_speed
  
```

---

---

**Algorithm 7** Commencement Emergency Response Simulation System (Part 3b: Pathfinding & Arrivals)
 

---

```

1: procedure PATHFINDToTARGET(current, target, speed)
2:   direction  $\leftarrow$  UnitVector(target – current)
3:   intended  $\leftarrow$  current + direction  $\times$  speed
4:   obstacles  $\leftarrow$  GetStadiumObstacles()
5:   for obstacle in obstacles do
6:     intersection  $\leftarrow$  LineRectangleIntersection(current, intended, obstacle)
7:     if intersection  $\neq$  null then
8:       intended  $\leftarrow$  ApplyObstacleSliding(intersection, obstacle, speed)
9:   return ClampToCanvas(intended)
10: procedure CHECKDESTINATIONARRIVALS(moving_agents)
11:   for agent in moving_agents do
12:     if Distance(agent.position, agent.target)  $\leq$  50 then
13:       if agent.destination is Exit then
14:         agent.state  $\leftarrow$  EXITED
15:       else
16:         agent.state  $\leftarrow$  WAITING
17:         RecordGroupArrival(agent.group, agent.destination, agent.id)
18: procedure CHECKGROUPARRIVALS(agents)
19:   resumed  $\leftarrow$   $\emptyset$ 
20:   for group in GetAllGroups(agents) do
21:     for destination in GetGroupDestinations(group) do
22:       chosen  $\leftarrow$  GetAgentsWhoChose(group, destination)
23:       arrived  $\leftarrow$  GetAgentsWhoArrived(group, destination)
24:       if chosen = arrived  $\wedge$  |chosen|  $>$  0 then
25:         for agent_id in arrived do
26:           agent  $\leftarrow$  GetAgent(agent_id)
27:           agent.state  $\leftarrow$  DISCUSSING
28:           resumed  $\leftarrow$  resumed  $\cup$  {agent}
29:         ClearDestinationTracking(group, destination)
30:   return resumed
  
```

---