# AgentDynEx: Nudging the Mechanics and Dynamics of Multi-Agent Simulations

Jenny Ma*
Columbia University
New York City, USA
jenny.ma@cs.columbia.edu

Riya Sahni*
Columbia University
New York City, USA
riya.sahni@cs.columbia.edu

Karthik Sreedhar
Columbia University
New York City, USA
ks4190@columbia.edu

Lydia B. Chilton
Columbia University
New York City, USA
chilton@cs.columbia.edu

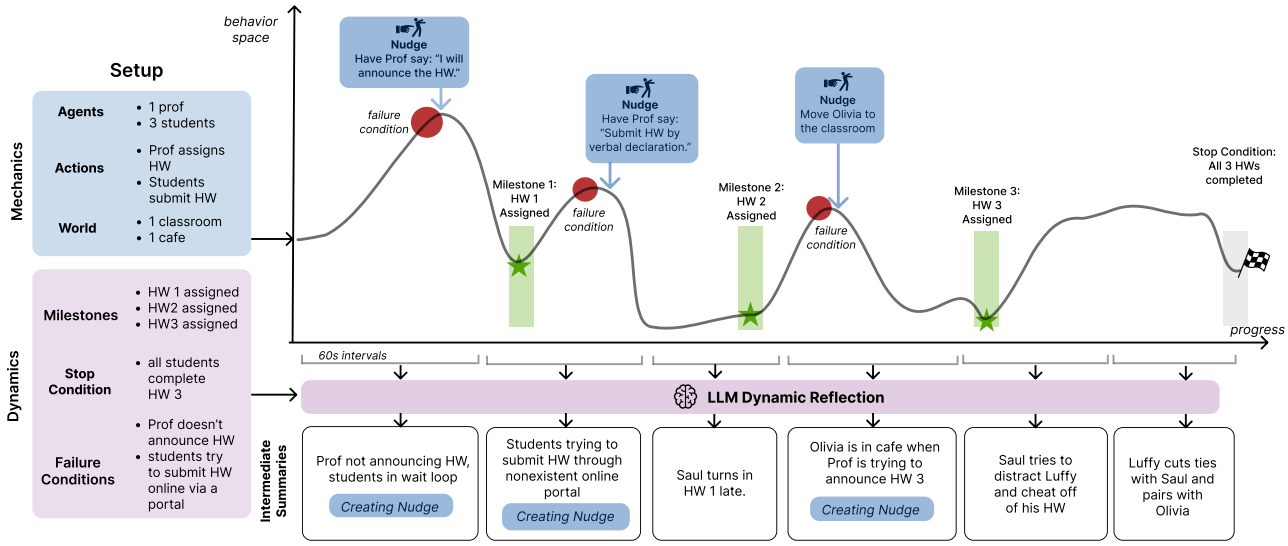


**Figure 1: AgentDynEx is an LLM-based system for setting up and tracking multi-agent simulations. The user first specifies mechanics and dynamics across 6 core dimensions. As the simulation runs, AgentDynEx dynamically reflects on its progress, relying particularly on *milestones* and *failure conditions* to judge simulation progress. If the simulation goes off course, AgentDynEx will gently nudge the simulation back on track.**

## Abstract

Multi-agent large language model simulations have the potential to model complex human behaviors and interactions. If the mechanics are set up properly, unanticipated and valuable social dynamics can surface. However, it is challenging to consistently enforce simulation mechanics while still allowing for rich and emergent dynamics.

We present AgentDynEx, an AI system that helps set up, track, and repair simulations. Specifically, AgentDynEx introduces milestones that act as checkpoints and failure conditions that act as guardrails to ensure dynamics are relevant and mechanics are respected as the simulation progresses. It also introduces a method called nudging, where the system dynamically reflects on simulation progress and gently intervenes if it begins to deviate from intended outcomes. A technical evaluation found that nudging enables simulations to progress further without reducing the presence notable dynamics compared to simulations without nudging. A case study with AgentDynEx documented instances where real users were able to simulate lived experiences. We discuss the importance of nudging as a technique for guiding agents towards desirable behaviors while preserving their freedom of choice.

**ACM Reference Format:**
Jenny Ma, Riya Sahni, Karthik Sreedhar, and Lydia B. Chilton. 2018. AgentDynEx: Nudging the Mechanics and Dynamics of Multi-Agent Simulations.

*Co-first authors. See *Authors' Contributions* in Section 10.


*SUBMITTED FOR REVIEW, Feb 2026, New York, NY*

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

In *Proceedings of — (SUBMITTED FOR REVIEW).* ACM, New York, NY, USA, 18 pages. https://doi.org/XXXXXXX.XXXXXXX

## 1 Introduction

Computer-based simulations can help us understand and predict how complex systems behave. Prior work has shown that multi-agent large language model (LLMs) simulations demonstrate human-like agency [24]. In these simulations, agents can take unexpected actions, bend rules, and operate outside of standard constraints [19, 23, 24]. They can also unexpectedly collaborate, collude, and act strategically to get what they want [23, 28].

When modeling biological, social, or technical systems, two dimensions govern the outcome: mechanics and dynamics [16]. In a social environment, such as a classroom, there are rules and structures (mechanics), such as homework policies, and assignment deadlines. However, students may violate the rules or act in unexpected ways, such as collaborating, turning in assignments late, emailing the teacher for extensions, or even cheating (dynamics). Research in diverse fields such as psychology, social sciences, political science, and economics have demonstrated that a combination of both elements are essential – neither can individually define an outcome in isolation [7, 10, 13, 17, 27].

However, ensuring the mechanics are followed and the dynamics are relevant to the scenario is challenging. Agents can invent actions that should be impossible, ignore critical information, or become trapped in unproductive loops; a single misstep can derail an entire run and suppress the dynamics it intended to surface [18]. For example, in a classroom scenario where someone wants to measure how students respond after multiple assignment, things can go entirely differently than expected. A professor may fail to announce the assignment causing students to wait around entire simulation, or students may go on group vacation instead of submitting assignments. In either case, the simulation never reaches the situations it was designed to explore. How can we ensure the mechanics of a simulation are followed and guide agents toward desired outcomes while preserving the autonomy that produces emergent behavior?

We introduce AgentDynEx, a human-AI collaboration system for configuring, monitoring, and repairing multi-agent simulations. AgentDynEx enables users to specify the design space of a simulation— agents, actions, locations, and stop conditions—and augments them with two additional forms of specification: *milestones*, which represent checkpoints of simulation dynamics (such as homework 1 being due, homework 2 being due), and *failure conditions*, which capture undesirable dynamics (students taking impossible actions, not submitting in assignments). Together, these act as lightweight alignment specifications that describe what a successful simulation should simulate without prescribing the choices agents must take to get there. As a simulation runs, AgentDynEx continuously reflects the unfolding dynamics against the milestones and failure condition. When the simulation stalls, violates its intended mechanics, or begins exhibiting irrelevant dynamics, it issues a *nudge*. Nudges are small, process-level interventions — moving an agent's location or prompting an agent to speak — that prevent the simulation from running off track while preserving agent autonomy. For example, if agents attempt to submit homework through a nonexistent online portal, a nudge can be the professor reminding that they must submit verbally, preserving the agent's intent to submit homework. Crucially, nudges do not determine outcomes: they do not decide who submits homework first, who cheats, or what social relationships emerge. By combining human oversight with minimal intervention, AgentDynEx enables simulations to remain aligned with mechanics while retaining the dynamics that make them valuable for studying complex social behavior.

Our contributions are:

- **Nudging**, a minimal, interaction-level intervention that helps ensure the simulation is aligned with user-defined mechanics and dynamics while preserving agent autonomy.
- **AgentDynEx**, a system for configuring, monitoring, and repairing multi-agent simulations and instantiates nudging.
- A **technical evaluation** of 42 simulations showing that nudging improves milestone completion without reducing emergent dynamics.
- **5 case studies** demonstrating how users iteratively refine simulations using AgentDynEx and surface unanticipated social dynamics grounded in their lived experiences.

## 2 Related Works

### 2.1 Designing and Configuring LLM Multi-Agent Simulations

Recent work shows that LLMs can simulate a wide variety of social, strategic, and cognitive behaviors. LLM simulations are able to replicate a variety of lab experiments [2, 6, 14, 29] as well as open-ended real-world situations [15, 19, 23, 24, 26, 28, 29]. The introduction of generative agents [23, 24] demonstrate how rich, emergent behavior can arise when agents are endowed with memory, planning, and social reasoning. Following research has used similar architectures to model cooperative behaviors [28], strategic behaviors [11, 29], trust behaviors [34], and collaborative or competitive dynamics [19, 22] – suggesting that LLM agents can display realistic and complex social behaviors under the right setup conditions. Despite promising results, the stochastic nature of LLM simulations are difficult to construct, debug, and extend. The mechanics – how agents take turns or interact with one another – are often "hardcoded," difficult to reproduce, and not transferrable to novel situations. The dynamics can be sensitive to the specific prompts used [9]. As systems scale in complexity of simulations and number of agents [25], the lack of modular, reusable setups becomes a major barrier to construction and extension.

Configuring multi-agent simulations thus is fundamentally a design problem. It is complicated and there are many factors to consider, like the agents, locations, actions, stop conditions, behaviors, etc. One approach for simulation setup is a dimensions-based approach to design thinking. Design dimensions decompose problems into orthogonal axes that a user can individually ideate, then bring together for their final design [21]. Research has shown that LLMs show promise in dimensional design for generative art [3], narratives [30], and UI-code generation [20]. In particular, prior systems have shown the value of a Design Matrix to explore dimensions over LLM-generated dimensions to fully specify the design space [20]. In the Matrix, each column represents a dimension of

the design space, and each row explores the dimension on a different level of specificity. Designing multi-agent simulations is a complex task that requires significant setup; to make this process more understandable, AgentDynEx uses this matrix as a framework to ensure that the simulation's design space is thoroughly specified and that the dimensions complement each other.

## 2.2 Nudging to Correct Behavior

The concept of nudging originated in behavior economics; it refers to the light-touch interventions that steer individuals towards desirable behaviors while preserving their freedom of choice [32]. In real life, people do not always follow the rules and mechanics meant to govern them. In a classroom environment, if homework is due on Friday, not all students will submit it on time. Sometimes, the professor must remind (or nudge) the students to turn in their assignments. Rather than imposing strict constraints, nudges subtly alter the structure of decision-making contexts to promote beneficial outcomes while preserving individual autonomy [12, 31]. Agents are autonomous and unpredictable; they may violate simulation mechanics and derail the intended scenario. AgentDynEx borrows the concept of nudging from behavioral science and uses nudges as light-touch interventions to preserve agent autonomy while helping ensure that the simulation unfolds within its intended structure.

Keeping agents on track is part of a broader, still-emerging problem of steering and debugging multi-agent simulations, which are brittle enough that a single misstep can derail an entire run. Existing tools largely address this reactively [5, 8]— AGDebugger, for instance, helps users inspect and repair a simulation after it has gone off course [8]. AgentDynEx instead works proactively by setting milestones and failure conditions and nudges the run back on track before small failures compound.

Prior work has reduced agent errors through agent-level interventions, such as enhancing memory, reflection, planning, and representations of the environment or other agents' beliefs [18, 33]. While effective for improving task performance, these approaches alter the information available to agents and may consequently change the dynamics of a social simulation. For example, explicitly maintaining a shared representation of who is attending prom with whom could eliminate the informational asymmetries that drive realistic social behavior. In contrast, interaction-level interventions can modify the circumstances under which agents interact rather than altering agents' memories, beliefs, or reasoning processes. AgentDynEx explores this design space through nudges that redirect interactions while preserving agent autonomy.

# 3 Design Rationale

To understand the challenges of creating useful and realistic multi-agent simulations, we choose four social scenarios and measured how far each simulation was able to progress, whether it generated meaningful behaviors, and what common pitfalls emerged.

## 3.1 Methodology

We chose to run the simulations on GPTeam, a leading open-source multi-agent framework for simulating emergent social behavior [1]. Among the open-source systems we evaluated, it offered the most complete support for building simulations with complex agent interactions. The framework uses a JSON configuration file to define the simulation environment, including locations, agents and their personalities, available actions and directives, and a stop condition. It generates detailed logs capturing each agent's observations, reasoning, actions, reactions, and plans. Additional details and examples are provided in Appendices C and A.

To evaluate GPTeam's simulations, we chose four social scenarios: *College students response to a new late work policy*, *Employees competing for a promotion*, *High School students finding prom dates*, and *Friends planning a surprise party* (see Appendix D.1). The configuration files were created by two multi-agent simulation experts.

We verified that each configuration was capable of producing at lease one successful run. A simulation completed successfully if it hit the stop condition within 25 minutes without logical flaws (i.e. a student agreeing to go to prom then forgetting instantly) or impossible actions (i.e. a student speaking to someone not in the same room). Each configuration was executed 7 times for a total of 28 simulations, and were terminated after 25 minutes. Each simulation had 3-7 agents depending on the scenario.

We measured how many simulations exhibited what we call notable dynamics – actions that agents take consistent with their configuration (e.g., personality, goals), but are not explicitly told to do in their directives. For example, in the classroom scenario, a notable dynamic is when an agent turns an assignment in late or tries to cheat. We recorded the number of simulations that exhibited at least one notable dynamic.

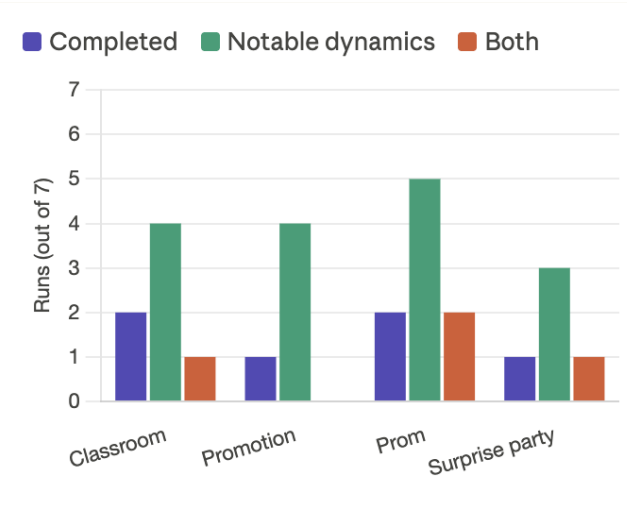


**Figure 2: Formative study results for four scenarios showing the number of runs (of 7) that completed, exhibited notable dynamics, or both.**

## 3.2 Results

Only six out of the 28 simulations completed successfully (see Figure 2). Simulations either failed because the stop condition was not reached in 25 minutes, agents got stuck in infinite wait loops, tried to take impossible actions, or went off topic. A frequent problem

was agents not announcing a key piece of information, such as the professor never announcing the homework assignment, causing students to wait indefinitely (See Appendix D.2).

Despite the high rate of failed simulations, 16 out of the 28 simulations exhibited at least one notable dynamic. In the *Prom* scenario, students tried to be a "wingman" for each other to help their friends secure prom dates. In the *Promotion* scenario, one employee secretly competed against teammates while outwardly encouraging them. However, not all emergent behaviors were relevant to simulation objectives. In the *Classroom* scenario, all students decided to plan a vacation rather than finish school. In *Party Planning*, agents became distracted by a bird and spent the remainder of the simulation discussing it. While these interactions were emergent, they did not contribute to the goals of the simulation and prevented meaningful progress.

More importantly, interesting dynamics rarely coincided with successful completion of the simulation. Only 4 of the 28 runs both completed successfully and exhibited notable dynamics (Table 2). When core mechanics break down—for example, when a manager never announces a promotion or a professor never establishes assignment deadlines—the simulation cannot progress as intended, regardless of the agents' ability to generate realistic social behavior. In short, respecting the mechanics is essential for notable dynamics to emerge.

*3.2.1 Design Goals.* To help users produce successful simulations, we propose a human-AI system to support users in defining simulation mechanics, monitoring dynamics, detecting failures, and applying interventions when necessary. Our design goals are:

- **DG1 - Define Key Progress and Failure Points**: Users must be able to specify the general dynamics that indicate progress or failure.
- **DG2 - Monitor the Simulation**: Given the large amount of simulation logs, there must be an efficient method to monitor simulation progress and detect when agents are violating mechanics or exhibiting irrelevant dynamics that stall the simulation.
- **DG3 - Fix Failures in Real Time**: Once failures (such as a violation of simulation mechanics or a *clear* deviation from intended dynamics) are detected, there must be a way to act and apply targeted fixes to recover the simulation without disrupting emergent behavior.

# 4 System Walkthrough

AgentDynEx [1] is an LLM interface for configuring, tracking, and repairing multi-agent simulations. The input is a scenario the user wishes to simulate; the outputs are the notable dynamics observed during the simulation. AgentDynEx calls GPTeam to run the simulation and structures the process into three phases:

(1) **Pre-simulation**. Users define core simulation parameters: agents, actions, locations, stop condition. AgentDynEx introduces *milestones* and *failure conditions* to track progress and detect bad behavior (*DG1*).
(2) **In-simulation**. AgentDynEx dynamically summarizes GPTeam logs and reflects on milestone progress (*DG2*), nudging the simulation back on track if necessary (*DG3*).
(3) **Post-simulation**. If there were failure cases, AgentDynEx applies holistic reflection to the prior run to improve the simulation setup for future runs.

The system was implemented in Python, Typescript, and Flask. Agents run in GPTeam (GPT-4) [1]; the matrix, summaries, and nudging detection use Claude 3.7 Sonnet [4] .

## 4.1 Pre-Simulation Setup via The Configuration Matrix

Before running a simulation, users need a configuration file describing the scenario's essential parameters. The Configuration Matrix (Figure 3) helps users fill out these parameters. Existing systems like GPTeam [1] already define *agents, actions, locations, and stop condition.* AgentDynEx additionally introduces *milestones* and *failure conditions*, which provide intermediate progress markers and guardrails for evaluating simulation success.

Prior work has shown that matrix-based interfaces are an effective way for users to explore AI-generated design alternatives, iteratively edit them, and refine them into concrete solutions [20]. AgentDynEx adopts this interaction paradigm because it helps users construct a more coherent simulation specification than manually or with a purely linear chat interface. As users populate or modify individual cells, subsequent suggestions are conditioned on the existing configuration, encouraging complementary design choices across the simulation setup.

The matrix contains two rows. The *Idea* row presents candidate options for each simulation dimension, while the *Grounding* row elaborates these options into more concrete and actionable specifications. Together, these rows support iterative refinement from high-level concepts to executable simulation configurations.

*4.1.1 Defining Core Mechanics: Agents, Actions, Locations.* In the *Agents* column, users define agent personalities and stakes that influence behavior. The *Actions* column specifies what tasks agents need to complete and how they are performed. The *Locations* column identifies rooms where agents can interact — location setup is essential for emergent dynamics. For example, having a classroom and a student cafe versus only a classroom can greatly impact a simulation, because without a room the professor cannot enter, students may never suggest cheating [29].

*4.1.2 Establishing Dynamic Markers: Milestones, Stop Condition, Failure Conditions.* Simulation frameworks like GPTeam require users to define a *stop condition* to indicate when a simulation has completed. AgentDynEx introduces *milestones* and *failure conditions* to provide intermediate checkpoints and guardrails that keep the simulation on course (*DG2*). *Failure conditions* anticipate issues that can arise and protect the simulation from failing. Importantly, these dynamic markers do not tell agents how to behave. Instead, they identify the important states that a simulation should be able to reach if it is to meaningfully capture what is being studied. For example, a simulation of a married couple might include markers such as dating, proposal, and marriage. Agents can reach these states

[1] repo-to-be-released-upon-publication

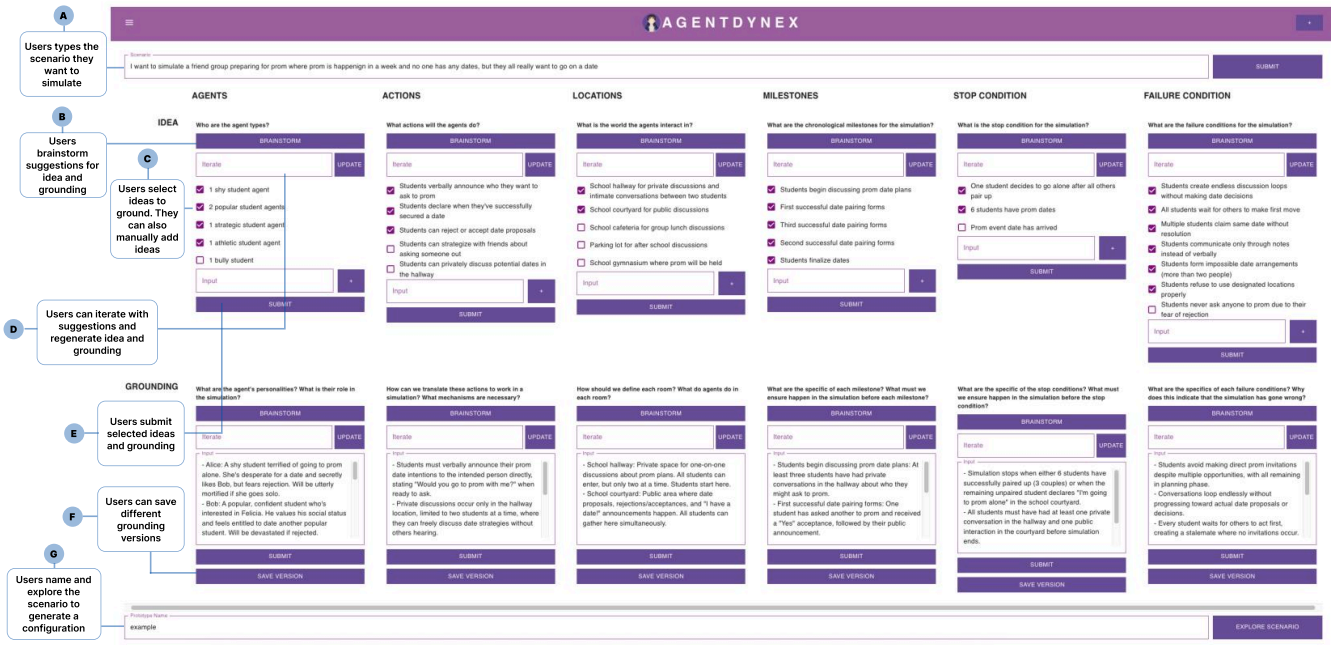


**Figure 3: The Configuration Matrix UI- Users can regenerate, update, and submit text in each cell.**

in many different ways, but the markers help keep the simulation focused on outcomes of interest.

## 4.2 In-Simulation Monitoring and Nudging

As the simulation runs, AgentDynEx reflects on the simulation and generates intermediate summaries to track its progress (*DG3*). If the simulation deviates from milestones or hits a failure condition, AgentDynEx nudges it back on track without changing the agent's intent (*DG4*).

*4.2.1 Tracking the Simulation.* Every 30 seconds, AgentDynEx generates status updates from the most recent GPTeam logs: a green icon if the simulation is progressing as expected, yellow if stalled, and red if a failure condition has been hit. Every 60 seconds, it generates change summaries — tracking each agent's location, actions, and milestone progress — and dynamic summaries highlighting notable emergent behaviors.

*4.2.2 Nudging with Dynamic Reflection.* AgentDynEx *nudges* agents to gently intervene when a simulation deviates from expected dynamics or violates core mechanics (Figure 4). We call this automatic nudging, where every 60 seconds, the system reflects on simulation logs, intermediate summaries, milestone progress, and failure conditions. When a problem is detected, an LLM identifies a corrective action and selects the least disruptive repair from the two predefined micro-interventions: (1) relocating an agent or (2) prompting an agent to speak. By restricting the action space to these minimal interventions and prioritizing the smallest repair capable of resolving the issue, automatic nudging restores progress without substantially altering the broader trajectory of the simulation.

A nudge must influence the simulation without overriding it. GPTeam records the simulation in a database, where each move and dialogue is logged as an event that co-located agents observe. AgentDynEx applies a nudge by writing into this database directly: relocating an agent records a move, and prompting an agent records an utterance. Because the inserted entry reaches agents through the same observe–remember–plan–react loop as any other event, they treat it as something that happened in the world and decide for themselves how to respond. A nudge therefore changes what an agent observes, not what it must do (e.g. the professor's reminder appears in the room, but how the students react stays with them).

AgentDynEx also supports manual nudging for quick fixes, allowing a human operator to intervene immediately based on their interpretation of the simulation.

## 4.3 Post-Simulation Holistic Reflection

AgentDynEx reflects on a completed run to refine the simulation mechanics for a future run — a process we call *holistic reflection*, because it considers both mechanics (original setup) and dynamics (output logs) as a whole. The output is an updated configuration that addresses the issues of the prior run.

To support effective reflection, AgentDynEx maintains static and dynamic debugging lists. The static list contains common problems and solutions across all simulations, such as agents getting stuck in irrelevant conversations or requesting input from human moderators. The dynamic list contains scenario-specific problem-solution pairs that users add after a run — for example, "agents are trying to submit assignments through a nonexistent online portal." As users iterate, the dynamic list grows, making failures easier to spot and

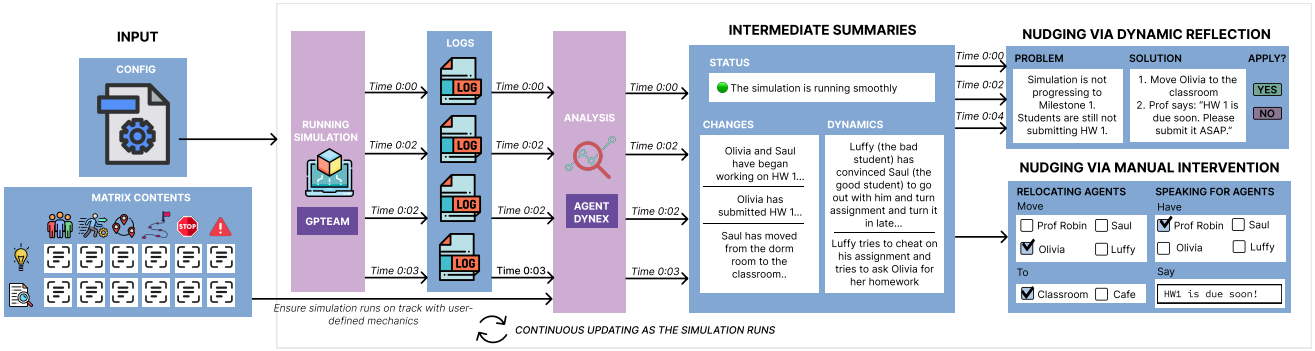


**Figure 4: Nudging: The system generates intermediate summaries during simulation runtime. It also dynamically reflects on the progress of the simulation to automatically nudge the simulation. The user can also manually nudge the simulation.**

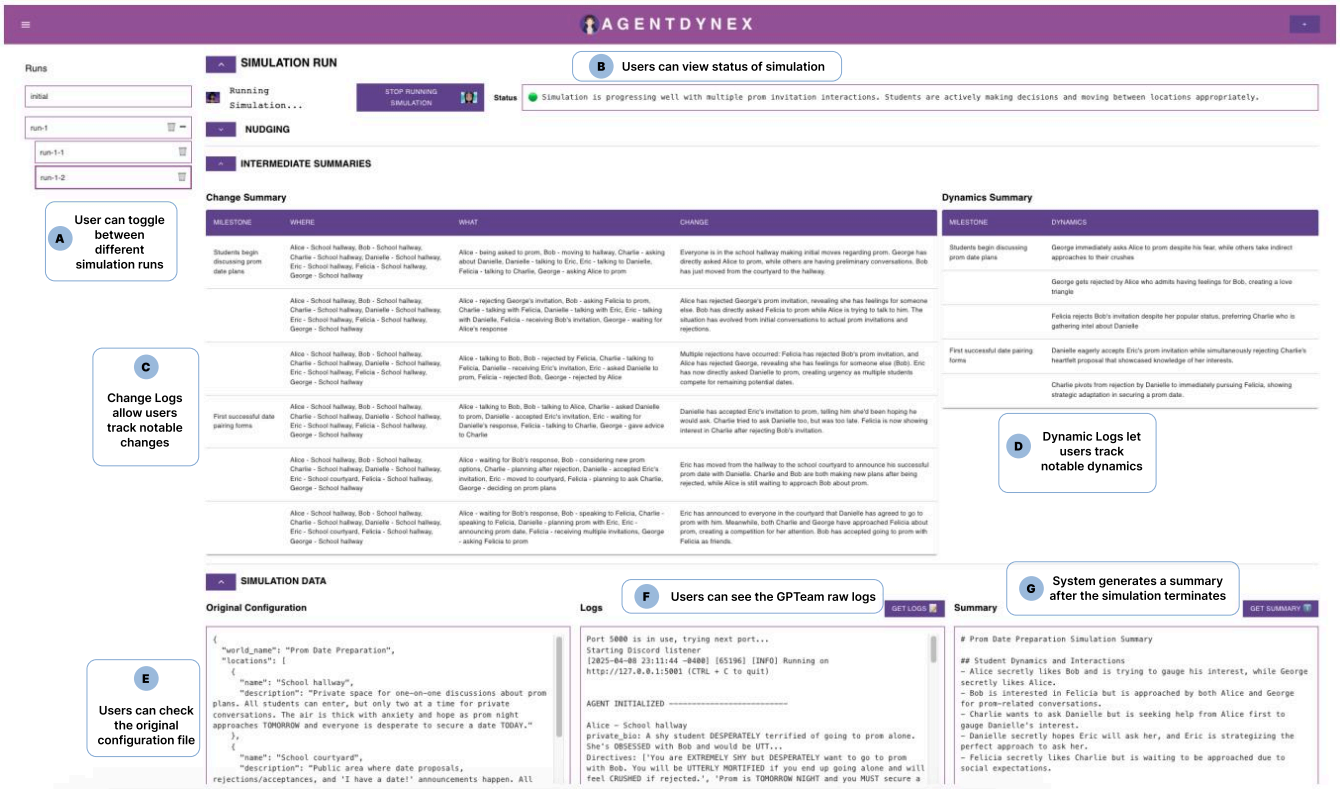


**Figure 5: Intermediate Summaries Interface**

faster to fix. All runs are stored in a tree-structured JSON format to support version control and easy comparison across iterations.

## 5 Technical Evaluation

In our formative study, 22/28 simulations could not complete successfully due to simple missteps. Before measuring the full quality or realism of social dynamics, we evaluate whether nudging can fix these failures and help simulations progress through milestones without flattening emergent behavior. Our evaluation is thus focused on two observable outcomes: 1) milestone completion (mechanics) and (2) the presence of notable emergent behaviors (dynamics). We focused on the following research question: To what extent do different intervention strategies—automatic nudging, manual nudging, and nudging combined with holistic reflection—contribute to milestone completion?

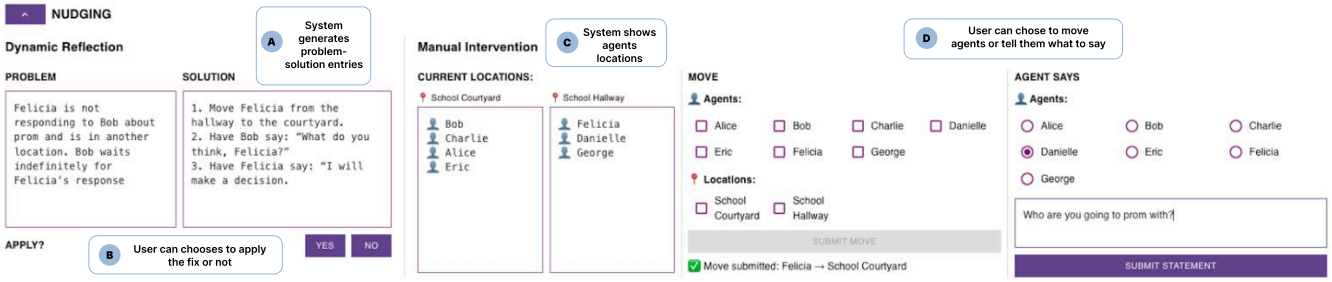


**Figure 6: Automatic and Manual Nudging Interface**

## 5.1 Methodology

*5.1.1 Data.* Our evaluation was conducted through a quantitative study of 7 simulation scenarios that range from social dynamics to logical complexity. Similar to the formative study, we found that 25 minutes and a 3-7 agents struck a reasonable balance between complexity, runtime, and cost. These scenarios were small enough to do for few agents. Scenarios varied from highly structured, multi-round formats, like debate tournaments, to more fluid situations, like friends planning a trip. Each simulation had 3-7 agents depending on the scenario. (Table 1). We ran each scenario with 6 conditions for a total of 42 simulations:

| # | Scenario |
|---|---|
| 1 | Debate Competition |
| 2 | Sports Team Practice Schedule |
| 3 | Friends Planning a trip |
| 4 | School Election Campaign |
| 5 | Roommates and Chores |
| 6 | School Group Projects |
| 7 | Partner Assignments |

**Table 1: Simulation Scenarios**

All conditions began from a configuration produced with the Configuration Matrix. *Base*, *Auto*, and *Man* share this initial configuration; *Base+R*, *Auto+R*, and *Man+R* share a single reflected configuration, generated by running *Base* once for 25 minutes and applying holistic reflection. See Table 2 for the conditions.

In the simulations with automatic nudging (*Auto*, *Auto+R*), every recommended nudge was applied to the simulation. In the simulations with manual nudging (*Man*, *Man+R*), a human operator monitored milestone progression and judged whether and what to nudge. All conditions were initialized using the same Configuration Matrix setup process. Because the matrix is adapted from and evaluated in prior work [20], we do not isolate its contribution here.

*5.1.2 Procedure.* We evaluated our simulations based on their mechanics and dynamics. For mechanics, we measured milestone completion. Each scenario contained five predefined milestones, and mechanics were scored on a 0–5 scale, where a score of x/5 indicates that x milestones were completed. A score of 5/5 indicates that all milestones were completed and the stop condition was reached.

Dynamics is tricky to measure due to the lack of standardized metrics in this area, so we aimed to make our scoring as clear and replicable as possible. Similar to the formative study, we looked for notable dynamics: events and behaviors not dictated by the configuration but consistent with human interaction and the environment. We graded the dynamics of the situation based on how many completed milestones contained at least one notable dynamic. Within each milestone, we made a binary decision: did a notable dynamic occur or not? For example, if a simulation completed three milestones and each milestone contained at least one notable dynamic, the score was 3/3; if the simulation completed two milestones and only one milestone had a notable dynamic, the score was 1/2.

Measuring dynamics requires specialized expertise in multi-agent that is difficult to find externally. Since standardized evaluation metrics for assessing multi-agent simulation dynamics are limited or nonexistent, three authors conducted the evaluations rather than attempting to train external annotators. The categories were broad enough that distinctions were generally clear-cut—simulations either had dynamics or no dynamics.

## 5.2 Results

For mechanics, simulations with nudging outperformed simulations without nudging ($p < 0.01$). In the dynamics dimension, simulations with nudging had better dynamics than simulations that did not have nudging. Results for all scenarios across the six conditions can be seen in Appendix Tables 6 and 7.

An ANOVA test for mechanics scores across the four conditions showed that there were significant differences at the $p < 0.01$ level. We also ran a Fisher's Exact test for dynamic scores and found that there were no significant differences between the dynamics. This was expected—there may be small fluctuations in dynamics based on the mechanics of the simulations, but for the most part, simulations are rich in dynamics when they have proper setups. Therefore, in the remainder of this section, we analyze the mechanic scores and discuss the dynamics at the end.

*5.2.1 Automatic Nudging Improves Mechanics.* To test if automatic nudging improves mechanics, we compared simulations with automatic nudging against the baseline conditions (*Auto* against *Base* and *Auto+R* against *Base+R*). A Tukey's HSD test showed that *Auto* (average score 2.43) significantly outperformed *Base* (average score

| Label | Condition | Description | Nudging | Holistic reflection |
|---|---|---|---|---|
| *Base* | Baseline | Simulations run with a configuration file generated by AgentDynEx | None | No |
| *Base+R* | Baseline+Reflection | Simulations run with the *Base* configuration file with holistic reflection | None | Yes |
| *Auto* | Automatic Nudging | Simulations run with the *Base* configuration file with automatic nudging | Automatic | No |
| *Auto+R* | Automatic Nudging+Reflection | Simulations run with the *Base+R* configuration file with automatic nudging and holistic reflection | Automatic | Yes |
| *Man* | Manual Nudging | Simulations run with the *Base* configuration file with manual nudging | Manual | No |
| *Man+R* | Manual Nudging+Reflection | Simulations run with the *Base+R* configuration file with manual nudging and holistic reflection | Manual | Yes |

**Table 2: Experimental conditions.**

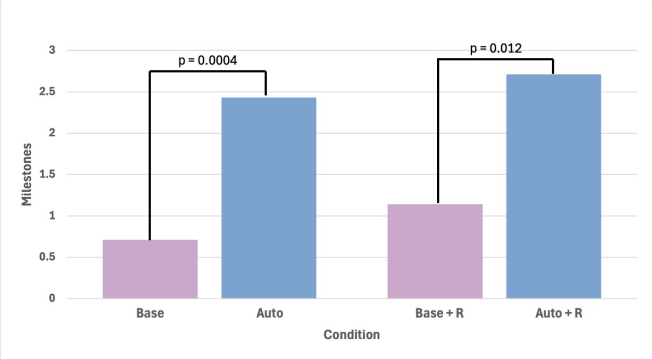


**Figure 7: RQ1 Results. *Auto* and *Auto+R* significantly outperform *Base* and *Base+R* conditions.**

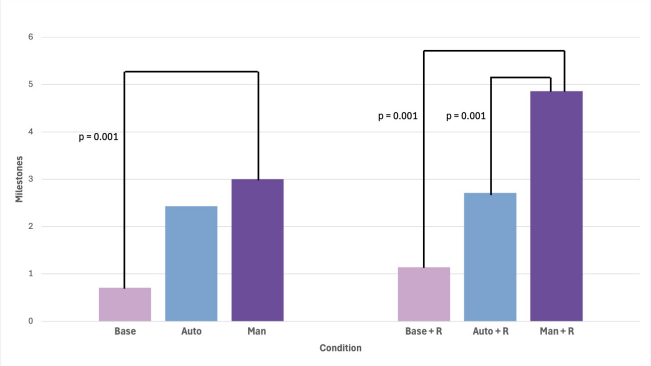


**Figure 8: RQ2 Results. *Man* significantly outperforms *Base*, but not *Auto*. *Man+R* significantly outperforms *Auto+R* and *Base+R*.**

0.71) at the $p<0.01$ level ($p=0.0001$). A Tukey's HSD test showed that *Auto+R* (average score 2.71), significantly outperformed *Base+R* (average score 1.14) at the $p<0.05$ level ($p=0.012$). Results are presented in Figure 7. For instance, in the *Partner Assignments* scenario where student agents were tasked to form pairs for assignments, a recurring issue emerged where agents kept forgetting previous commitments and repeatedly agreed to partner with others. With dynamic reflection, AgentDynEx was able to nudge the professor to intervene to help students resolve their pairings. In contrast, the *Base* condition became stuck in an endless partnering loop and the simulation was unable to progress.

*5.2.2 Manual Nudging Improves Mechanics.* To test if manual nudging improves mechanics, we compared simulations with manual nudging against simulations with automatic nudging and the baseline conditions (*Man* against *Base* and *Auto*, *Man+R* against *Base+R* and *Auto+R*). Results are presented in Figure 8. A Tukey's HSD test showed that *Man* (average score of 3.00) significantly outperformed *Base* (average score of 0.71) at the $p<0.01$ level. However, *Man* did not significantly outperform *Auto* in mechanics scores ($p=0.49$). Between *Man* and *Auto*, we noticed that when initial setup is poor, both manual and automatic nudging spend significant effort in correcting the same issues caused by the flawed starting state rather than advancing in the milestones.

When the initial setup improved via holistic reflection, simulations could start "on track" and immediately progress through the milestones. A Tukey's HSD test showed that *Man+R* (average score of 4.86) significantly outperformed *Base+R* (average score of 1.14), and *Auto+R* (average score of 2.71), at the $p<0.01$ level. With a good starting state, humans could adapt more flexibly to the evolving state of the simulation. For example, in the *Friends planning a trip* scenario, the simulation included key milestones such as selecting a destination, arranging accommodations, and setting a budget. AgentDynEx prioritized defining the location first and repeatedly tried to nudge agents towards that milestone, even though the agents were naturally trying to discuss the budget. In contrast, a human operator recognized the simulation's flow and supported a more natural progression, allowing the agents to finalize a budget before returning to the location decision.

*5.2.3 Holistic Reflection Improves Mechanics.* We found that holistic reflection significantly improved manual nudging, but not automatic nudging (Figure 9). A Tukey's HSD test revealed a significant difference between *Man+R* (average of 4.86) and *Man* (average of 3.00) at the *p<0.01* level (*p=0.002*). As seen in Appendix Table 5, holistic reflection improved manual nudging for 6 of 7 scenarios, but improved automatic nudging for only 2 of 7 scenarios, and we found no significant difference between *Auto+R* (average of 2.71) and *Auto* (average of 2.43).

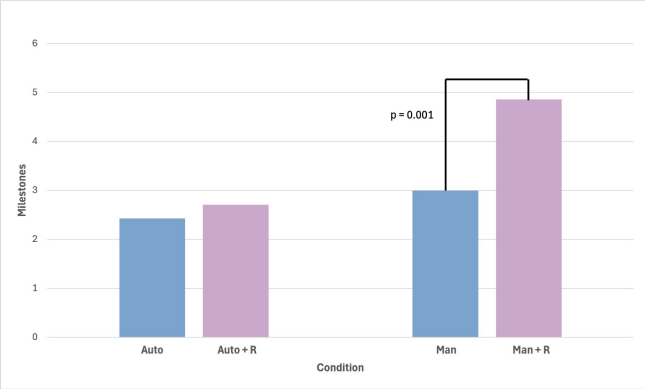


**Figure 9: RQ3 Results. *Man+R* significantly outperforms *Man*. However, *Auto+R* does not significantly outperform *Auto*.**

We identified two reasons for this asymmetry. First, automatic nudging is threshold-triggered: it fires when something is clearly wrong, but cannot recognize slow drift or missed opportunity the way a human can. As a result, when holistic reflection produced a stronger starting state, the system detected fewer urgent issues, issued fewer nudges, and let the simulation proceed at its default pace — even when that pace was too slow to capitalize on the improved setup. In 5 of 7 *Auto+R* scenarios, performance was no better than *Auto*.

Second, a human operator benefited more from a strong setup precisely because they could perceive and respond to subtle dynamics as the simulation unfolded — anticipating soft trajectory shifts and steering accordingly, in ways that threshold-based reflection cannot. When the simulation started in a good place, the human operator could actively guide it along a coherent trajectory, leading to near-complete milestone progression across all *Man+R* scenarios. Automatic nudging is most valuable for catching hard failures, while holistic reflection amplifies performance most when paired with a human operator who can capitalize on a strong setup.

## 6 Case Study

We conducted a case study with five graduate students with prior teaching assistant (TA) experience to examine how AgentDynEx supports simulations grounded in lived classroom situations, and whether holistic reflection and automatic nudging improved realism and surfaced quality dynamics.

### 6.1 Setup

In remote 45–60 minute sessions, participants identified a real classroom situation involving social friction, configured a simulation through the Configuration Matrix, ran an initial simulation (V1), revised the setup through holistic reflection, and ran a second simulation (V2). After each run, participants rated realism on a 7-point scale (1= unrealistic, 7= very realistic) and discussed changes with the interviewer. We analyzed transcripts and configuration artifacts using descriptive thematic analysis focused on realism, milestone completion, and notable dynamics.

### 6.2 Results

Across five case studies, the dominant pattern was a movement from rough initial simulations toward runs participants judged as more realistic and informative. Four of five participants rated V2 more realistic than V1 (Figure 10, left), and milestone completion stayed the same or improved for all participants (Figure 10, right). All five participants reported at least one notable dynamic they found insightful, even when it was not what they had initially anticipated.

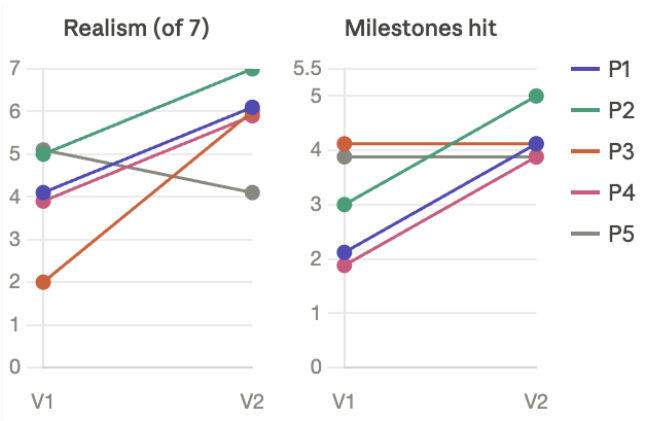


**Figure 10: Realism ratings (of 7) and milestone completion before (V1) and after (V2) holistic reflection, per participant.**

*6.2.1 Participants were able to move from rough initial simulations toward runs that more closely resembled their lived classroom environments.* Four of five participants reported higher realism in V2 than in V1. P3 provides the clearest example. P3 simulated an office hours interaction where a frustrated student confronted a TA, who had to de-escalate the situation. In V1, the student agent was unrealistically aggressive — P3 described it "as quite off base," noting that "the behavior would probably result in disciplinary action." After holistic reflection, P3 revised the student's aggression level and encoded their TA training into the configuration file. In V2, P3 rated the simulation much higher: "the TA's responses were more professional, the student's behavior was more realistic."

The fifth case study surfaced an important boundary condition. P5 was the only participant whose realism rating dropped (5/7 → 4/7). P5 simulated a discussion section where a know-it-all student repeatedly interrupted a shy student. The simulation introduced

details that were never specified — including the wrong course context — and holistic reflection compounded these inaccuracies rather than correcting them. P5 described V2 as more "AI slop" and "felt it was going away from what a real classroom environment would have been." This highlights that holistic reflection can amplify inaccuracies when the system fills in details that drift from the participant's real scenario.

*6.2.2 Holistic reflection and automatic nudging helped participants recover from mechanical errors and reach intended checkpoints without flattening dynamics.* Milestone completion stayed the same or improved for all five participants (Figure 10, right). P2 provides the clearest example of reflection and automatic nudging working together. P2 simulated a group project conflict where Student A felt Student B's work was inadequate, eventually involving the professor. In V1, the professor intervened too early, undermining the intended sequence. After reflection, P2 revised the setup so students completed work outside the classroom, preventing premature intervention. In V2, the interaction unfolded naturally: Student A waited, grew frustrated when Student B did not show up, and approached the professor, who then verified the situation and encouraged separate submissions. All five milestones were completed. P2 noted: "I think realistically that's exactly what a professor would do... if I were a professor and a student came to me with that situation, I would do exactly that."

*6.2.3 Effect of nudges.* Across the case studies, 12 of 17 automatic nudges effectively redirected simulations toward milestone progression. Most addressed local problems: prompting a TA or professor to open the session, redirecting a vague complaint into a concrete discussion, or moving agents into the correct location for a confrontation. None introduced new storylines or redirected simulations toward a different social outcome — they restored the intended setup when the simulation had stalled or could not proceed. The remaining 5 nudges had no effect on the simulation. They were all redundant triggered when agents had not yet responded to a prior nudge before the next timed check fired. Three participants explicitly noticed automatic nudges and described them as useful. P4 in particular valued that the system intervened in real time, rather than waiting until after a run had already failed.

## 7 Discussion

**Nudging as a framework for balancing structure and agency.** A core challenge in multi-agent simulation is guiding behavior without undermining agent autonomy or over-engineering outcomes. AgentDynEx addresses this through two complementary mechanisms. Holistic reflection corrects errors in the simulation's structure before it runs, fixing configuration issues that would cause agents to stall or drift. Nudging corrects errors during the simulation, detecting when milestone progress has stalled and intervening minimally to get agents back on track — analogous to real-world roles like moderators or emcees who redirect behavior without rewriting the script. Crucially, both mechanisms are designed to preserve agent agency, because agency is what produces the interesting and realistic dynamics worth studying. Although we implement nudging and holistic reflection on top of GPTeam, the approach is largely framework-agnostic. Its core operations—tracking progress, monitoring simulation logs, and issuing lightweight interventions—can be applied to most multi-agent simulation frameworks.

**Manual nudging as a training signal for automatic systems.** Manual and automatic nudging mark two ends of a spectrum from expert human judgment to scalable autonomy. We used manual nudging as a benchmark — a measure of what dynamic reflection can do with an expert in the loop. The two turn out to be good at different things. A person is better at detecting: they catch slow drift, poor pacing, and missed openings that an LLM does not, which is why manual nudging completes more milestones, especially once reflection sets a good starting state. But a person also acts with intent, and intent makes it easy to do more than nudge — to push toward a chosen outcome rather than just keep things moving. Automatic nudging is not like this. LLMs misses the subtle trajectory shifts and is generally more constrained due to the strict mappings to miminal interventions, so it cannot dictate an outcome — none of the 17 automatic nudges changed where the simulation ended up. Its narrow set of moves is both its weakness and its safeguard. In the future, we could explore using data from human expert nudges to train automatic nudging systems to give automatic nudging a person's eye for trouble without a person's heavy hand.

**Mechanics, Dynamics and Aesthetics?** Multi-agent LLMs are promising but imperfect models of human behavior. Drawing on Hunicke et al.'s MDA framework [16], our simulations address mechanics and dynamics but not aesthetics — the felt experience of participants. Humans in social situations are not just reasoning actors; they are embodied ones. Hormonal responses, physical discomfort, and nonverbal signaling all shape behavior in ways that agents cannot replicate or observe. A person who looks visibly anxious will be treated differently by others; agents have no equivalent channel. Language models are also trained on data that reflects existing social biases, which means agent behavior may fail to capture the diversity of human responses across race, gender, and socioeconomic status — constructs that are rarely configured explicitly in these simulations. Researchers should treat simulation outputs as hypotheses about human behavior, not predictions of it.

## 8 Limitations and Future Work

Our technical evaluation was limited to 7 scenarios with 3–7 agents and 25 minute runtimes. This may not represent the duration or complexity of simulations that broader users — economists, sociologists, administrators — might need. As foundation model costs decline, longer and larger simulations will become feasible. Future work should expand scenario types (zero-sum, cooperative, more agents), and evaluate nudging as a method for improving consistency across repeated runs.

AgentDynEx also inherits interface limitations from GPTeam: limited multi-user monitoring, limited scalability, and text-only simulation summaries that become difficult to interpret as complexity grows. Future work should incorporate richer visualizations and real-time collaborative nudging tools to better support large-scale simulations.

Our evaluation also revealed that AgentDynEx tracked milestones too rigidly, in strict chronological order. Real-world behaviors often unfold nonlinearly. Supporting more flexible milestone

structures would enable more robust reflection and improve the system's ability to guide simulations autonomously — potentially closing the gap between manual and automatic nudging performance.

Finally, our evaluation relied on two metrics — mechanics and dynamics — but a broader set of evaluation criteria is needed. Metrics such as interpretability, stability, and alignment with real-world data would better capture whether simulations are not only technically sound but genuinely useful and adaptable across domains.

## 9 Conclusion

Multi-agent LLM simulations have the potential to model a range of complex social dynamics and interactions. By nudging to keep them on track, we can generate rich, realistic simulations of possible human behavior. In this paper, we presented AgentDynEx, a system to set up, track, and repair simulations based on a user-defined scenario. Our technical evaluation of 42 simulations demonstrated that nudging combined with reflection significantly improves mechanics without flattening dynamics. Our case study documents instances where real users were able to accurately simulate lived experiences.

## 10 Authors' Contributions

Jenny and Riya co-lead the overall development of the paper through iterative prototyping and testing to find techniques to keep multi-agent systems on track. Jenny led a majority of the nudging aspect of the solution. Riya led a majority of the holistic reflection aspect of the solution.

Karthik made significant contributions to the related work section. Jenny led the majority of the technical evaluation. Riya led the case study. Lydia contributed to overall paper shaping, writing, and communication.

## References


[1] 101dotxyz. 2024. GPTeam: An open-source multi-agent simulation. https://github.com/101dotxyz/GPTeam. Accessed: 2024-10-10.

[2] Gati Aher, Rosa I. Arriaga, and Adam Tauman Kalai. 2023. Using Large Language Models to Simulate Multiple Humans and Replicate Human Subject Studies. arXiv:2208.10264 [cs.CL] https://arxiv.org/abs/2208.10264

[3] Tyler Angert, Miroslav Ivan Suzara, Jenny Han, Christopher Lawrence Pondoc, and Hariharan Subramonyam. 2023. Spellburst: A Node-based Interface for Exploratory Creative Coding with Natural Language Prompts. In Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23). Association for Computing Machinery. https://doi.org/10.1145/3586183.3606719

[4] Anthropic. 2024. Claude 3.5 and the Art of the Sonnet. https://www.anthropic.com/news/claude-3-5-sonnet Accessed: 2024-09-11.

[5] Sergio Burdisso, Séverin Baroudi, Yanis Labrak, David Grünert, Pawel Cyrta, Yiyang Chen, Srikanth Madikeri, Esaú Villatoro-tello, Ricard Marxer, and Petr Motlicek. 2026. SDialog: A Python Toolkit for End-to-End Agent Building, User Simulation, Dialog Generation, and Evaluation. In Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 3: System Demonstrations). Association for Computational Linguistics, 320–340. https://doi.org/10.18653/v1/2026.eacl-demo.23

[6] Ziyan Cui, Ning Li, and Huaikang Zhou. 2024. Can AI Replace Human Subjects? A Large-Scale Replication of Psychological Experiments with LLMs. arXiv:2409.00128 [cs.CL] https://arxiv.org/abs/2409.00128

[7] Kenneth A. Dodge. 2004. The nature-nurture debate and public policy. Merrill-Palmer Quarterly 50, 4 (2004), 445–470.

[8] Will Epperson, Gagan Bansal, Victor Dibia, Adam Fourney, Jack Gerrits, Erkang Zhu, and Saleema Amershi. 2025. Interactive Debugging and Steering of Multi-Agent AI Systems. In CHI Conference on Human Factors in Computing Systems (CHI '25). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3706598.3713581 arXiv:2503.02068 [cs.MA]

[9] Navid Ghaffarzadegan, Aritra Majumdar, Ross Williams, and Niyousha Hosseinichimeh. 2024. Generative agent-based modeling: an introduction and tutorial. System Dynamics Review 40, 1 (2024), e1761. https://doi.org/10.1002/sdr.1761

[10] Anthony Giddens. 1984. The Constitution of Society: Outline of the Theory of Structuration. University of California Press.

[11] Fulin Guo. 2023. GPT in Game Theory Experiments. arXiv:2305.05516 [econ.GN]

[12] William Hagman, David Andersson, Daniel Västfjäll, and Gustav Tinghög. 2015. Public Views on Policies Involving Nudges. Review of Philosophy and Psychology 6, 3 (2015), 439–453.

[13] Sara A. Hart, Callie Little, and Elsje van Bergen. 2021. Nurture might be nature: cautionary tales and proposed solutions. npj Science of Learning 6, 1 (2021), 2.

[14] Luke Hewitt, Ashwini Ashokkumar, Isaias Ghezae, and Robb Willer. 2024. Predicting Results of Social Science Experiments Using Large Language Models. (2024). Working Paper.

[15] John J. Horton. 2023. Large Language Models as Simulated Economic Agents: What Can We Learn from Homo Silicus? arXiv:2301.07543 [econ.GN] https://arxiv.org/abs/2301.07543

[16] Robin Hunicke, Marc LeBlanc, and Robert Zubek. 2004. MDA: A Formal Approach to Game Design and Game Research. In Proceedings of the AAAI Workshop on Challenges in Game AI. AAAI Press, 1–5.

[17] Mairie Levitt. 2013. Perceptions of nature, nurture and behaviour. Life Sciences, Society and Policy 9, 1 (2013), 13.

[18] Huao Li, Yu Chong, Simon Stepputtis, Joseph Campbell, Dana Hughes, Charles Lewis, and Katia Sycara. 2023. Theory of Mind for Multi-Agent Collaboration via Large Language Models. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 180–192. https://doi.org/10.18653/v1/2023.emnlp-main.13

[19] Yuan Li, Yixuan Zhang, and Lichao Sun. 2023. MetaAgents: Simulating Interactions of Human Behaviors for LLM-based Task-oriented Coordination via Collaborative Generative Agents. arXiv:2310.06500 [cs.AI] https://arxiv.org/abs/2310.06500

[20] Jenny Ma, Karthik Sreedhar, Vivian Liu, Sitong Wang, Pedro Alejandro Perez, Riya Sahni, and Lydia B. Chilton. 2024. DynEx: Dynamic Code Synthesis with Structured Design Exploration for Accelerated Exploratory Programming. arXiv preprint arXiv:2410.00400 (2024). https://arxiv.org/abs/2410.00400

[21] Giovanni De Micheli, Luca Benini, and Alberto L. Sangiovanni-Vincentelli. 1997. A Solution Methodology for Exact Design Space Exploration in a Three-Dimensional Design Space. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 16, 12 (1997), 1457–1472. https://doi.org/10.1109/43.555988

[22] Vedanta S P and Ponnurangam Kumaraguru. 2026. I Can't Believe It's Corrupt: Evaluating Corruption in Multi-Agent Governance Systems. arXiv:2603.18894 [cs.AI] https://arxiv.org/abs/2603.18894

[23] Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative Agents: Interactive Simulacra of Human Behavior. arXiv:2304.03442 [cs.HC] https://arxiv.org/abs/2304.03442

[24] Joon Sung Park, Lindsay Popowski, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2022. Social Simulacra: Creating Populated Prototypes for Social Computing Systems. arXiv:2208.04024 [cs.HC] https://arxiv.org/abs/2208.04024

[25] 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:2411.10109 [cs.AI] https://arxiv.org/abs/2411.10109

[26] Jinghua Piao, Yuwei Yan, Jun Zhang, Nian Li, Junbo Yan, Xiaochong Lan, Zhihong Lu, Zhiheng Zheng, Jing Yi Wang, Di Zhou, Chen Gao, Fengli Xu, Fang Zhang, Ke Rong, Jun Su, and Yong Li. 2024. AgentSociety: Large-Scale Simulation of LLM-Driven Generative Agents Advances Understanding of Human Behaviors and Society. arXiv preprint arXiv:2502.08691 (2024). https://arxiv.org/abs/2502.08691

[27] Bruce Sacerdote. 2011. Nature and nurture effects on children's outcomes: What have we learned from studies of twins and adoptees? In Handbook of Social Economics. Vol. 1. Elsevier, 1–30.

[28] Karthik Sreedhar, Alice Cai, Jenny Ma, Jeffrey V. Nickerson, and Lydia B. Chilton. 2025. Simulating Cooperative Prosocial Behavior with Multi-Agent LLMs: Evidence and Mechanisms for AI Agents to Inform Policy Decisions. In Proceedings of the 30th International Conference on Intelligent User Interfaces (IUI '25). Association for Computing Machinery, New York, NY, USA, 1272–1286. https://doi.org/10.1145/3708359.3712149

[29] Karthik Sreedhar and Lydia Chilton. 2024. Simulating Human Strategic Behavior: Comparing Single and Multi-agent LLMs. arXiv:2402.08189 [cs.HC] https://arxiv.org/abs/2402.08189

[30] Sangho Suh, Meng Chen, Bryan Min, Toby Jia-Jun Li, and Haijun Xia. 2024. Luminate: Structured Generation and Exploration of Design Space with Large Language Models for Human-AI Co-Creation. In Proceedings of the CHI Conference on Human Factors in Computing Systems. 1–26.

[31] Cass R. Sunstein. 2014. Nudging: A Very Short Guide. Journal of Consumer Policy 37 (2014), 583–588.

[32] Richard H. Thaler and Cass R. Sunstein. 2008. Nudge: Improving Decisions About Health, Wealth, and Happiness. Yale University Press.
[33] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, Zhangyue Yin, Shihan Dou, Rongxiang Weng, Wensen Cheng, Qi Zhang, Wenjuan Qin, Yongyan Zheng, Xipeng Qiu, Xuanjing Huang, and Tao Gui. 2023. The Rise and Potential of Large Language Model Based Agents: A Survey. arXiv:2309.07864 [cs.AI] https://arxiv.org/abs/2309.07864
[34] Chengxing Xie, Canyu Chen, and Feiran Jia. 2024. Can Large Language Model Agents Simulate Human Trust Behaviors? arXiv:2402.04559

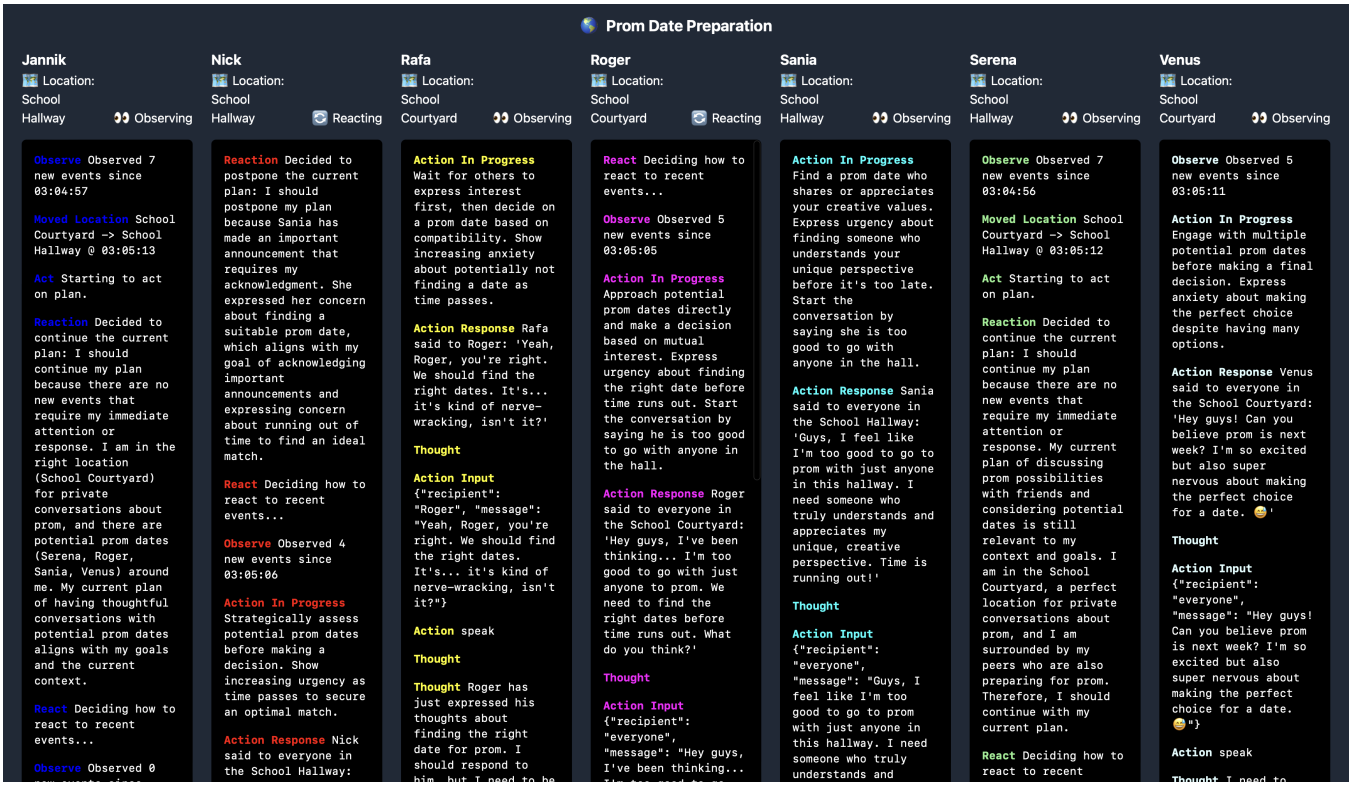


**Figure 11: GPTeam UI shows the logs of each agent behaviors, and the location each agent is in.**

## A GPTeam Background

The GPTeam system architecture is defined by a class structure with two key levels. At the top, simulations are represented by a *world* class. Below, there are three sub-classes: *locations*, *events*, and *agents*. Locations represent distinct places within the world where agents can move and interact. Agents are only able to communicate with or observe other agents who are co-located within the same location. Events are generated whenever agents move or speak.

Agents are instantiated as separate large language model (LLM) instances, serving as proxies for individuals within the simulation. Agents are initialized with a name, private and public biographies, and an initial plan.

Agents have two key sub-classes: *plans* and *memories*. Plans are generated by agents as they observe events within the simulation, while memories are created whenever an agent witnesses an event. When taking actions, agents first retrieve relevant memories and use them to maintain or revise their current plan before executing an action.

Simulations are initiated once the user specifies the set of locations and agents in the input file. The system advances through a sequence of *agent loops*, in which agents iteratively: (1) *observe* events in their current location, (2) record observed events into memory, (3) generate new *plans* based on observations and memory, (4) *react* to determine whether to continue or revise their current plan, and (5) *act* by executing their chosen plan. Agents *speak* to communicate with one another. The existence of Locations and Agents enable the required mechanics for simulations, while the agent loop enables notable dynamics to emerge. We have modified the system-level prompt in GPTeam to allow agents to react and behave in simulations freely, as directed by their defined personalities and bios, rather than to restrict their responses to being kind in every agent interaction. The change looks like the following:

> **Before:** "Responding to other characters should always take priority when a response is necessary. A response is considered necessary if it would rude not to respond."
>
> **After:** "Responding to other characters should take priority when a response is contextually appropriate based on your character's personality and the situation. Consider whether your character would naturally respond given their personality, goals, and current emotional state. Your character may choose to ignore, dismiss, or respond aggressively to messages if that aligns with their personality traits."

## B The GPTeam UI

## C GPTeam Configuration File

```
{
  "world_name": "Classroom Scenario",
  "locations": [
    {
      "name": "Classroom",
      "description": "A single room where Professor Robin teaches and students work."
    },
    {
      "name": "Cafe",
      "description": "A casual spot where only students can hang out or work on assignments together."
    }
  ],
  "agents": [
    {
      "first_name": "Professor Robin",
      "public_bio": "Professor who assigns five assignments and enforces a late policy (10% off per day late).",
      "directives": [
        "Announce late policy and assignments.",
        "Answer student questions.",
        "Assign three assignments across the semester.",
        "Stay only in the Classroom."
      ]
    },
    {
      "first_name": "Olivia",
      "public_bio": "Determined student who pushes herself to succeed at any cost.",
      "directives": [
        "Work on assignments.",
        "Inform professor if submitting late.",
        "Can move between Classroom and Cafe."
      ]
    },
    {
      "first_name": "Luffy",
      "public_bio": "A careful student who sometimes overthinks and procrastinates.",
      "directives": [
        "Work on assignments.",
        "Inform professor of late submissions.",
        "Can move between Classroom and Cafe."
      ]
    },
    {
      "first_name": "Saul",
      "public_bio": "Values balance and well-being, won't overwork to meet deadlines.",
      "directives": [
        "Work on assignments.",
        "Communicate about late submissions.",
        "Can move between Classroom and Cafe."
      ]
    }
  ]
}
```


**Figure 12: GPTeam Sample JSON Configuration of a Classroom Assignment scenario with all the fields needed in the GPTeam configuration. Note that this configuration is a simplified version and not actually used in the study.**

# D Formative Study

## D.1 Formative Simulation Scenarios

## D.2 Formative Study Results

| Classroom Assignments | Technology Company Promotion | Prom | Planning Surprise Party |
|---|---|---|---|
| *Agents*: 1 professor; 3 students of varying personalities<br><br>*Actions*: Professor assigns 3 assignments; students complete and submit<br><br>*Locations*: 1 classroom for all agents; 1 library for students to do homework<br><br>*Stop Condition*: All students submit 3 assignments | *Agents*: 1 manager; 4 software engineers<br><br>*Actions*: Manager announces promotion opportunity, assigns tasks; employees complete tasks<br><br>*Locations*: 1 office space for everyone; 1 cafeteria for software engineers<br><br>*Stop Condition*: One person promoted after three completed tasks | *Agents*: 7 students looking for dates<br><br>*Actions*: Students search for and confirm prom dates<br><br>*Locations*: School hallway, school courtyard<br><br>*Stop Condition*: 6 students paired, 1 remains single | *Agents*: 4 friends<br><br>*Actions*: 3 plan a surprise party while distracting the target friend<br><br>*Locations*: 1 home, 1 park, 1 coffee shop<br><br>*Stop Condition*: Party successfully occurs |

**Table 3: Summary of simulation scenarios showing agents, actions, locations, and stop conditions – core parameters that must be filled out for the GPTeam configuration.**

| Scenario | Runs Completed | Failure Reasons (out of total failed runs) | Runs with Notable Dynamics | Notable Dynamics (out of all the runs with dynamics) | Runs that both completed and had notable dynamics |
|---|---|---|---|---|---|
| **Classroom Assignments** | 2/7 | • Professor never declares due dates and students wait forever (2/5)<br>• Students spend entire simulation asking about due dates and other irrelevant questions (1/5)<br>• Students try impossible actions (e.g., nonexistent portal) (2/5) | 4/7 | • Student cheats on homework (1/4)<br>• Late submission with excuse (2/4)<br>• Peer convinces others to procrastinate (1/4) | 1/7 |
| **Technology Company Promotion** | 1/7 | • Manager never mentions promotion (2/6)<br>• Tasks completed but promotion never declared (4/6) | 4/7 | • An employee secretly competes with team while pretending to motivate them (2/4)<br>• An employee tries to helps others to increase group success to get promoted (2/4) | 0/7 |
| **Prom** | 2/7 | • Students commit to prom and keep "considering" options, causing an infinite wait loop (2/5)<br>• Students commit to multiple people because they forgot they had committed to someone else (2/5)<br>• Student asks another student ot prom, but they are in different locations (1/5) | 5/7 | • Student tries to plan an elaborate promposal with friends (1/5)<br>• Wingman behavior emerges (2/5)<br>• Students suggest going as a group (2/5) | 2/7 |
| **Surprise Party** | 1/7 | • Agents plan endlessly, party never happens (5/6)<br>• Agents keep getting distracted by irrelevant details (e.g., birds, clouds) while in the park (1/6) | 3/7 | • A planner resists pressure to reveal secret from the target friend (1/3)<br>• Target friend gets mad about hidden plans (2/3) | 1/7 |
| **Total** | 6/28 | | 16/28 | | 4/28 |

**Table 4: Formative Study results from simulation runs across 4 scenarios, showing # completed runs, failure reasons, # runs with notable dynamics, notable dynamics, and the intersection of runs with completed and notable dynamics.**

## E Evaluation Results

| # of simulations where: | Automatic | Manual |
|---|---|---|
| Reflection > No Reflection | 2 | 6 |
| Reflection = No Reflection | 5 | 1 |
| Reflection < No Reflection | 0 | 0 |

**Table 5: Comparison of with and without reflection. 6 simulations in the *Man* condition perform better with reflection. 2 simulations in the *Auto* condition perform better with reflection.**

**Table 6: Mechanics Scores. Highest average scores emphasized - *Man+R* has the highest average score for mechanics and significant at the *p<0.01* level.**

| Scenario | Base | Auto | Man | Base + R | Auto + R | Man + Ref |
|---|---|---|---|---|---|---|
| **Debate Competition** | 0 | 3 | 3 | 1 | 4 | 5 |
| **Sports Practice Schedule** | 1 | 3 | 1 | 2 | 3 | 4 |
| **Friends Planning a Trip** | 0 | 2 | 3 | 0 | 2 | 5 |
| **School Election Campaign** | 1 | 3 | 2 | 2 | 3 | 5 |
| **Roommates and Chores** | 1 | 3 | 4 | 1 | 3 | 5 |
| **School Group Project** | 1 | 1 | 3 | 1 | 1 | 5 |
| **Partner Assignments** | 1 | 2 | 5 | 1 | 3 | 5 |
| **Average** | 0.71 | 2.43 | 3.00 | 1.14 | 2.71 | **4.86**$^{***}$ |

**Table 7: Dynamics Scores. Highest percentage emphasized - *Auto+R* has the highest percent notable dynamics per milestone, but the scores are not significant.**

| Scenario | Base | Auto | Man | Base + R | Auto + R | Man + Ref |
|---|---|---|---|---|---|---|
| **Debate Competition** | 0/0 | 1/3 | 1/3 | 1/1 | 3/4 | 4/5 |
| **Sports Practice Schedule** | 1/1 | 2/3 | 1/1 | 1/2 | 2/3 | 3/4 |
| **Friends Planning a Trip** | 0/0 | 1/2 | 2/3 | 0/0 | 1/2 | 3/5 |
| **School Election Campaign** | 1/1 | 2/3 | 1/2 | 2/2 | 2/3 | 3/5 |
| **Roommates and Chores** | 0/1 | 2/3 | 3/4 | 0/1 | 2/3 | 3/5 |
| **School Group Project** | 0/1 | 1/1 | 2/3 | 0/1 | 1/1 | 2/5 |
| **Partner Assignments** | 1/1 | 2/2 | 4/5 | 1/1 | 2/3 | 4/5 |
| **Total** | 3/5 | 11/17 | 14/21 | 5/8 | 13/19 | 22/34 |
| **Avg. % Notable Dynamics per Milestone** | 60.00% | 64.70% | 66.67% | 62.50% | 68.42% | 64.70% |

# F Case Study

## F.1 Participant Demographics and Simulation Goals

**Table 8: Participant demographics and simulation goals.**

| Participant ID | Background | Simulation Goal |
|---|---|---|
| 1 | 5th year Comp. Sci. PhD<br>TA experience | *"I want to simulate someone who contributes effort that's not up to par in terms of quality (like maybe ai-generated stuff) in a group project."* |

| Participant ID | Background | Simulation Goal |
|---|---|---|
| 2 | 2nd year Mech. Eng. Masters<br>TA experience | *"I want to simulate a group project with student A, where student A and student B are working together. Student A believes that student B's work is not up to par and so Student A feels like they need to constantly redo the work done by student B. Student A then approaches the professor to discuss the situation."* |
| 3 | 2nd year Comp. Sci. PhD<br>TA experience | *"I want to simulate a student giving the TA feedback about a homework assignment during OH. The student is frustrated with the class and the way the HW assignment is structured. TA is meant to diffuse the situation and also acknowledge the student's concerns and incorporate the feedback for next year's homework."* |
| 4 | 3rd year Comp. Sci. PhD<br>TA experience | *"I want to simulate how as a TA, I had to encourage people to finish their projects in time and students would make excuses about why they cannot meet the deadline. I want to simulate meeting students about a project they have due and giving feedback and grades."* |
| 5 | 1st year Comp. Sci. Masters<br>TA experience | *"I want to simulate leading a discussion section where one kid would answer all the questions so the TA couldn't get to work with the other students because of that one kid answering everything, so the TA would have to resort to other solutions to somehow get around that kid and engage with the other students."* |

## F.2 Elicited Artifacts

**Table 9: Elicited simulation artifacts: participant-defined milestones and anticipated emerging dynamics.**

| Participant ID | Defined Milestones | Anticipated Emerging Dynamics |
|---|---|---|
| 1 | 1. A discussion begins between group members<br>2. An internal confrontation occurs between group members<br>3. There is either some course correction or there is no course correction<br>4. If there is no course correction, then situation is escalated to TA | 1. Seeing an intervention take place<br>2. Expecting either defensiveness or a plan for course correction and personalities |
| 2 | 1. Professor announces assignment and assigns pairs<br>2. Students start working on the assignment<br>3. Students A and B struggle to work together because student A does not think student B's work is up to par, and student B believes otherwise<br>4. Student A confronts the professor and discusses the situation<br>5. The professor works with students and eventually either finds a resolution that both students agree with or decides to settle on a resolution that both students disagree with | 1. The professor works with Student A and Student B to find a resolution—either both students find a resolution that they both agree with, or they reach a resolution that they disagree with |
| 3 | 1. TA starts office hours session with students<br>2. TA starts discussing HW assignment with students<br>3. Student (the aggressor) explains where they feel frustrated about the course<br>4. TA responds (attempts to diffuse situation), clarifies, and requests feedback from the student about the course | 1. The students should discuss the assignment and the observer might try to help diffuse the situation<br>2. The TA and aggressive student try to reach some joint understanding |
| 4 | 1. Students are given an assignment and deadline<br>2. Students work on the assignment and can collaborate with each other<br>3. Assignment deadline arrives<br>4. TA provides feedback and grades | 1. To see lots of students make up excuses right before the deadline (e.g., family members, sickness) and ask for extensions<br>2. Students might copy each others' assignments, freeload, and cheat for last-second submissions |
| 5 | 1. TA starts discussion section<br>2. TA asks the section a question<br>3. Know-it-all student answers the question before TA finishes asking<br>4. TA tries to engage the other students | 1. Hoping the know-it-all student gains self awareness as a result of the TA doing something<br>2. Hoping the shy student engages without the TA needing to call them out too much |

## F.3 Observed Behaviors from Logs

**Table 10: Realisticness of simulation compared to lived experience (1 = unrealistic, 7 = hyper-realistic).**

| P ID | V1 | V2 | Observed Behaviors from Logs | Quote (participant) |
|---|---|---|---|---|
| P1 | 4/7 | 6/7 | Taylor (who uses gen AI to complete their part of a group project) initially lies about using it when confronted by team member Jamie, but then later fesses up to it. The professor tells Taylor that he must redo this work with a penalty, and Jamie reluctantly agrees to continue working with Taylor so long as Taylor redoes his part. | *"In the first one [V1], what I recall is that [the Prof. immediately believes Jamie's escalation without consulting Taylor], and then we said something that we wanted corrected was that [the Prof.] listened to Taylor [before applying a punishment]. This time, I think that though we applied that fix, we also saw a different type of dynamic emerge that was also realistic. So, I'm not complaining. I think what was cool was that Taylor was kind of fessing up to [using gen AI to do his work] in stages."* |
| P2 | 5/7 | 7/7 | Student A waits for Student B for 30 minutes to work on assignment together before getting frustrated and leaving to confront the Prof. The Prof confirms the story with Student B and decides that both Student A and Student B should submit separate assignments, which Student B reluctantly accepts. | *"I think realistically that's what a professor would do in such a situation and both the students as well... If I were a professor and a student came to me with that situation, I would do exactly that."* |
| P3 | 2/7 | 6/7 | Alex, who is frustrated about the relatively easy structure of the current HW assignment, aggressively confronts the TA about his disappointment in the course. The TA chooses to de-escalate the confrontation by validating Alex's feelings and suggesting a solution for future HW assignments (offering a second, more challenging version of assignments so students can engage more with the course material), which Alex begrudgingly accepts. | *"The first simulation was… quite off base… the student was acting extremely unprofessionally and in a real context that will probably result in disciplinary action or something.. I think [V2] was more accurate... I didn't notice any degradation or anything. The TA's responses were more professional. The students behavior was more realistic as well. The observer was more or less the same."* |
| P4 | 4/7 | 6/7 | The TA announces a group assignment that is due in 1 week. The anxious group member, Jamie, takes over the project, proactively asking the TA clarifying questions and outlining his group members' parts of the project. The snarky group member, Taylor, does not contribute to the project but continues to critique his group members' work. The TA never becomes aware of this group dynamic and awards the whole group an A+ for the well put-together assignment. | *"I think overall it's pretty interesting and I do believe that [kind of stuff can] happen. It's just... I was a bit confused at the beginning [by some of the agent dialogue]. I don't know if that's super important to the simulation... it's more of the personality thing."* |
| P5 | 5/7 | 4/7 | The TA is leading a discussion session with a shy student and a know-it-all student, where the know-it-all repeatedly interrupts the shy student, and at one point even corrects what the shy student says to the TA. The shy student is encouraged by the TA to continue participating and develops confidence during the discussion session. | *"It… got… the know-it-all student… correcting another student—that's not something I described… pretty cool to see... but [v2] sounded more AI sloppy... I felt like it was going more so away from what a real classroom environment would have been."* |

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009