# CollabSim: A CSCW-Grounded Methodology for Investigating Collaborative Competence of LLM Agents through Controlled Multi-Agent Experiments

**Jiaju Chen**
Northeastern University

**Bo Sun**
Northeastern University

**Yuxuan Lu**
Northeastern University

**Yun Wang**
Microsoft Research Asia

**Dakuo Wang**
Northeastern University

**Bingsheng Yao***
Northeastern University

## Abstract

Multi-agent systems (MAS) built on large language models have shown growing promise, with their effectiveness resting on agents' ability to coordinate through text-based channels much as human teams do. Yet recent study suggests that MAS often falter not because agents lack individual task-solving ability, but because they lack collaborative competence: the capacity to establish common ground, maintain shared task understanding, balance individual and collective incentives, and repair misalignment as interaction unfolds. Decades of research in Computer-Supported Cooperative Work have characterized these requirements for human teams coordinating under constrained communication, yet existing MAS evaluations focus mainly on task outcomes or single-agent proficiency in reasoning, planning, and tool use. To enable a systematic analysis of agents' collaborative competence in MAS, we introduce CollabSim, a configurable simulation framework that combines a theory-grounded definition of collaborative capabilities, controlled manipulation of interaction conditions, and action-level probing of agents' internal states. Experiments across four LLMs show that CollabSim can capture condition effects, separate model performance patterns, and reveal task-dependent effects of agent design.

## 1 Introduction

A growing body of work has explored multi-agent systems (MAS) for complex task solving, with applications in software engineering (He et al., 2025), scientific research (Gottweis et al., 2025), and web automation (Zhou et al., 2023). As these systems tackle increasingly complex tasks, their effectiveness increasingly rests on the collaborative mechanism among agents, where multiple agents with distinct roles perform individual tasks, exchange information through text-based channels, and coordinate toward shared goals (Chen et al., 2025; Zou et al., 2025). The ability of individual agents to collaborate effectively with their partners thus becomes as consequential as their ability to solve tasks independently (Wang et al., 2024a).

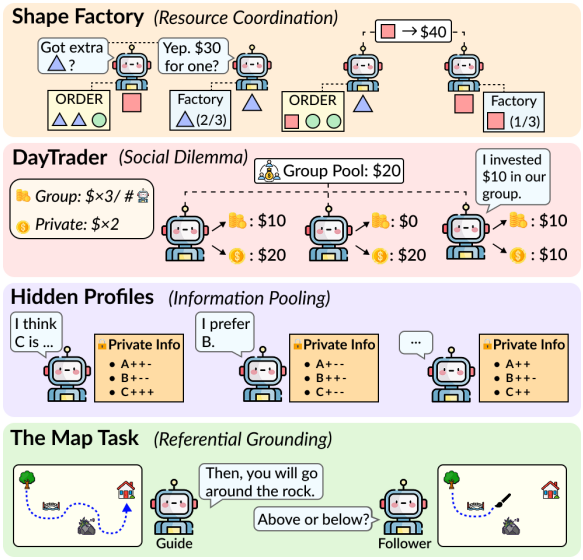


Figure 1: Illustrations of the four multi-agent experiments instantiated in CollabSim: Shape Factory (Bos et al., 2004), DayTrader (Bos et al., 2002), Hidden Profile (Stasser and Titus, 1985), and The Map Task (Anderson et al., 1991).

A parallel line of evidence, however, suggests that MAS often fail not because individual agents lack task-solving capacity, but because they struggle to coordinate during collaboration (Cemri et al., 2026; Li et al., 2026). Agents have been shown to misread partners' states and intentions (Mu et al., 2026), lose track of shared task progress as context accumulates (Pappu et al., 2026), fail to repair misalignment (Yadav et al., 2026), and struggle to balance individual and collective incentives (Tewolde et al., 2026). We refer to this broad capacity as agents' **collaborative competence**.

However, existing evaluation methodologies are

* Corresponding Author: b.yao@northeastern.edu.

not designed to diagnose process-level failures in agents' collaborative competence. Current MAS benchmarks primarily measure outcome-centric task-solving performance (Liu et al., 2024; Zhu et al., 2025; Sun et al., 2025) or assess individual agents' proficiency in reasoning, planning, and tool use (Qin et al., 2023; Yao et al., 2024). Yet, these evaluations are insufficient for diagnosing the coordination breakdowns identified above, since those failures arise from agents interaction process rather than from any single agents' capability deficit.

Notably, the coordination failures described above closely parallel long-standing problems studied in Computer-Supported Cooperative Work (CSCW) research on distributed human teams, where collaborators similarly coordinate through structured text-based channels without co-presence. Research in this tradition has identified the core competencies required for effective remote collaboration, including common ground maintenance (Clark and Brennan, 1991), shared task understanding (Cannon-Bowers et al., 1993), and misalignment repair (Gutwin and Greenberg, 2002). Prior work has also shown that collaborative competencies vary with interaction conditions like communication bandwidth, information visibility, and team structure (Bos et al., 2002; Daft and Lengel, 1986). To assess these competencies under controlled conditions, CSCW researchers developed experimental paradigms that isolate specific dimensions of collaborative competence (Bos et al., 2002, 2004; Stasser and Titus, 1985; Anderson et al., 1991), thereby providing a validated methodology for evaluating the same competencies in LLM agents.

Drawing on this methodology, we introduce `CollabSim`[1], a configurable simulation framework for systematically assessing agents' collaborative competence under controlled interaction conditions. `CollabSim` allows researchers to vary task constraints such as communication bandwidth, information visibility, and team size, so that the effects of specific interaction conditions on collaborative behavior can be examined. It also includes a probing module that elicits each agent's reported mental model awareness of the task state, partner intentions, and own reasoning after every action, enabling analysis of agents' internal collaborative states beyond observable behavior. To demonstrate the framework's coverage, we implement four CSCW-inspired collaborative tasks that reflect the process-level challenges discussed above: resource coordination under cost asymmetry (Shape Factory; Bos et al. 2004), social-dilemma negotiation between individual and collective incentives (DayTrader; Bos et al. 2002), distributed information pooling under information asymmetry (Hidden Profile; Stasser and Titus 1985), and referential grounding through language-only spatial communication (Map Task; Anderson et al. 1991).

We validate `CollabSim` across the four tasks under systematically varied interaction conditions, comparing two agent designs (persona-based and Collaboration-Theory-Informed) across four model backbones (Qwen3.6-35B-A3B, Llama-4-Maverick-17B-128E-Instruct-FP8, GPT-5.5, and Claude 4.6 Sonnet). The results suggest that `CollabSim` provides stable and interpretable measurements: condition manipulations shift collaboration metrics in expected directions across models, the same task settings separate proprietary from open-source models, and cross-task evaluation reveals task-dependent effects of agent design.

## 2 Related Work

### 2.1 LLM-Based Multi-Agent Systems

Early LLM-based multi-agent systems (LLM-MAS) (Wu et al., 2024; Qian et al., 2024; Hong et al., 2024) often instantiate agents with predefined personas and communication protocols. More recent work (Chen et al., 2024, 2025; Zhang et al., 2025) has explored dynamic multi-agent architectures, where task decomposition, role assignment (Wang et al., 2025), agent dependencies, and execution workflows (Yang et al., 2026) can be adapted based on user input. These systems have been applied across various domains (Fang et al., 2026; Zhou et al., 2026), including software engineering (He et al., 2025), web automation (Drouin et al., 2024), scientific reasoning (Gottweis et al., 2025), etc. These LLM-MAS primarily treat agent-agent collaboration as a design strategy for improving task completion (Tran et al., 2025; Cemri et al., 2026) instead of assessing or improving agents' collaborativeness for effective coordination.

### 2.2 Frameworks for Agent Collaboration

Recent work has developed a variety of frameworks for evaluating and simulating multi-agent collaboration. The first line introduces task-based evaluation benchmarks. These frameworks place multiple

[1]Code is available at `https://github.com/neuhai/CollabSim`.

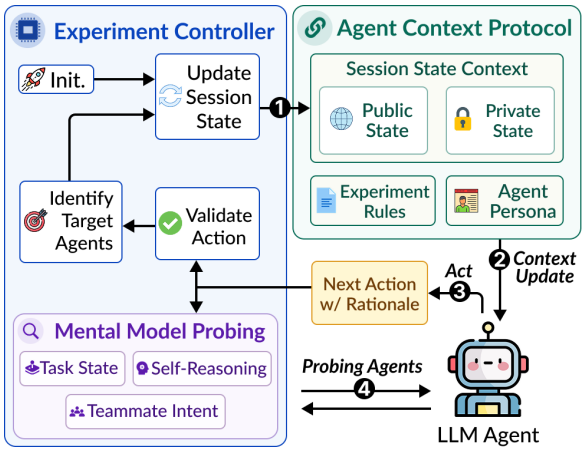


Figure 2: Architecture of `CollabSim`. The Controller manages the action validation, state update, and mental model probing for each agent.

LLM agents in collaborative settings and assess their performance across different scenarios and interaction protocols (Yu et al., 2025; Sun et al., 2025; Orogat et al., 2026; Tewolde et al., 2026). MultiAgentBench (Zhu et al., 2025), for example, covers a range of cooperative tasks and measures team-level outcomes under varied coordination structures. Another line of work focuses on role-play-based social simulation environments, where LLM agents are instantiated with personas, social goals, and relationship contexts to simulate dyadic (Wang et al., 2024b, 2026) or population-level interactions (Park et al., 2024; Yang et al., 2024; Piao et al., 2025). For instance, SOTOPIA (Zhou et al., 2024) provides an open-ended social interaction environment for evaluating agents' social intelligence.

These frameworks demonstrate that LLM agents can coordinate, communicate, and behave socially in structured settings. However, existing benchmark frameworks (Wang et al., 2024c; Sun et al., 2025) primarily evaluate agents' task-solving outcomes and overlook the effectiveness of agent collaboration and the coordination among agents. Social simulation frameworks (Zhou et al., 2024; Piao et al., 2025) expose rich agent behavioral traces, but they often treat collaboration conditions as fixed design choices rather than experimental variables.

## 3 `CollabSim`

We present `CollabSim`, a theory-grounded simulation framework for systematic investigation of multi-agent collaborative competence across classic CSCW experimental paradigms. We describe the system architecture that governs how agents perceive, act, and are probed (Sec. 3.1), the configurable interaction conditions that serve as experimental variables (Sec. 4.3.1), the probing module that captures agents' internal states (Sec. 3.3), and the four collaborative tasks that cover distinct dimensions of collaborative competence (Sec. 3.4).

### 3.1 System Architecture

As shown in Figure 2, `CollabSim` organizes its architecture into two layers: an **interaction layer** that governs how agents perceive the task environment, and a **control layer** that orchestrates experiment configuration, execution, and evaluation. Figure 2 illustrates how these layers interact during one complete agent action cycle. Formally, a collaboration experiment is defined as a tuple

$$\mathcal{E} = (\mathcal{A}, \mathcal{S}, \{\mathcal{X}_i\}_{i=1}^n, \{\mathcal{O}_i\}_{i=1}^n, T),$$

where $\mathcal{A} = \{a_1, \ldots, a_n\}$ is the set of agents, $\mathcal{S}$ is the state space, $\mathcal{X}_i$ is the action space available to agent $a_i$, $\mathcal{O}_i$ is the observation space of agent $a_i$, and $T$ is the termination condition. At each turn $t$, the task maintains a shared state $s^t \in \mathcal{S}$. Each agent $a_i$ receives an observation $o_i^t \in \mathcal{O}_i$, selects an action $x_i^t \in \mathcal{X}_i$, and the state transitions to $s^{t+1}$ according to task-specific dynamics.

**Agent Context Protocol.** The Agent Context Protocol defines the interface through which agents perceive the task environment and act. In the beginning, the agent context protocol is initialized with the experiment rules and agent persona. At each turn $t$, the system constructs each agent's observation $o_i^t$ by combining two components: a *public update*, which reflects the current shared task state $s^t$ filtered by information-visibility rules, and a *private state*, which carries agent-specific facts such as remaining production capacity or assigned candidate materials. The agent then selects an action $x_i^t$ from the action space $\mathcal{X}_i$. By standardizing how context is constructed and delivered, the protocol allows diverse tasks to share a unified agent interface while preserving task-specific semantics.

**Experiment Controller** The Experiment Controller manages simulation execution through a turn-based loop driven by a `.yml` configuration file, which specifies agent roles, LLM backend settings, interaction conditions, task parameters, and probing questions. At initialization, the Controller parses the configuration, initializes the state $\mathcal{S}^0$, the action spaces $\mathcal{X}_i$, and each agent $a_i$ with its assigned roles and prompts. At each turn $t$, as shown

in Figure 2, the Controller constructs $o_i^t$ via the Agent Context Protocol, queries the agent for the next action; once the agent responds, the Controller validates the returned action $x_i^t$ against $\mathcal{X}_i$, applies its effects to transition the state to $s^{t+1}$ for targeted agents, and triggers the Probing Module before turn $t+1$. The loop continues until the termination condition $T$ is satisfied. All observations, actions, and probe responses are logged in structured JSON for downstream analysis. Implementation details are reported in Appendix A.

### 3.2 Configurable Interaction Conditions

CSCW research has established that collaboration dynamics are shaped by the richness of communication channels (Daft and Lengel, 1986), the visibility of teammates' states (Gutwin and Greenberg, 2002), and team structure (Bos et al., 2004). `CollabSim` translates these factors into three categories of configurable experimental variables:

**Communication Bandwidth** controls the frequency and length of messages agents can exchange, paralleling media richness manipulations in CSCW experiments (Daft and Lengel, 1986; Bos et al., 2002).

**Information Visibility** controls what shared state information is accessible beyond private observations; for example, an awareness dashboard in Shape Factory displays teammates' balance and order progress (Gutwin and Greenberg, 2002), and a canvas visibility condition in Map Task lets the guide observe the follower's drawing progress.

**Group Size** varies the number of agents per session (e.g., 4/8/10 in Shape Factory, 3/6/9 in DayTrader) to examine how coordination demands scale. Table 1 summarizes the conditions available for each task.

### 3.3 Probing Module

Observable actions reveal what agents do during collaboration but provide limited insight into the reasoning that motivates those actions. Drawing on shared mental model theory (Cannon-Bowers et al., 1993) and Clark's common ground framework (Clark and Brennan, 1991), the Probing Module queries each agent after every action along three dimensions: **perceived task state, perceived teammate intents, and self-reasoning.** Question templates are customized for each task (full templates are in Appendix D). We derive two quantitative measures from the responses: a self-reported confidence score capturing each agent's perceived certainty about shared task understanding, and a pairwise response similarity computed via SBERT-encoded (Reimers and Gurevych, 2019) cosine similarity, which captures the degree to which agents converge on shared representations over time.

### 3.4 Collaborative Tasks

As shown in Figure 1, `CollabSim` includes four tasks, each targeting a distinct dimension of collaborative competence and each drawn from an established CSCW or social science experimental paradigm. These tasks capture common process-level mechanisms in agent teams, including coordinating interdependent actions, managing individual-team incentive conflicts, integrating distributed information, and establishing common ground under asymmetric information.

Shape Factory (Bos et al., 2004) targets resource coordination under interdependence. In this task, each agent produces a specialty shape at a lower cost while receiving orders only for other shapes, so agents must negotiate trades to complete orders and maximize earnings. Each specialty is assigned to two agents; thus, the experiment creates a market-like setting that assesses agents' capacity to establish efficient coordination structures through communication and negotiation.

DayTrader (Bos et al., 2002) targets incentive management in a social dilemma. In each round, agents allocate funds between private investments (individual returns only) and group investments (higher collective returns shared equally). The task is well-suited to studying how agents balance a tension between self-interest and group interests.

Hidden Profile (Stasser and Titus, 1985) targets information pooling under asymmetric knowledge. In this task, decision-relevant information is distributed such that no agent alone can identify the optimal candidate, and shared information supports a suboptimal choice. Agents must exchange unique facts and integrate them into a revised collective decision, which maps to the common ground maintenance process (Clark and Brennan, 1991).

Map Task (Anderson et al., 1991) targets referential grounding under spatial asymmetry. In this task, a guide holding a map with landmarks and a route must instruct a follower (who has a map with landmarks only) to reproduce the route through text-based chat. This asymmetric-information dyadic task supports studying how agents establish shared references, request clarification, and repair misunderstandings during spatial communication.

| Interaction Conditions | Shape Factory | DayTrader | Hidden Profile | Map Task |
|---|---|---|---|---|
| BASELINE | ✓ | ✓ | ✓ | ✓ |
| Communication Bandwidth | ✓ | ✓ | ✓ | ✓ |
| Group Size | ✓($n=4,8,10$) | ✓($n=3,6,9$) | – | – |
| Information Visibility | ✓ | – | – | ✓ |
| **Total Conditions** | 4 | 3 | 2 | 3 |

Table 1: Interaction conditions enabled for each task.

## 4 Benchmark Experiments

To validate whether `CollabSim` can reveal meaningful differences in agents' collaborative behaviors across varying tasks, conditions, model backbones, and agent designs, we conducted large-scale experiments across the four tasks, evaluating (1) agents powered by different LLMs, (2) agents with different collaborative designs (persona-based vs. theory-informed), and (3) systematically manipulated interaction conditions (communication bandwidth, group size, and information visibility).

### 4.1 Experiment Setup

**Models and Agent Designs.** We evaluate four LLMs spanning both open-source (Qwen3.6-35B and Llama-4)[2] and proprietary (GPT-5.5 and Claude 4.6 Sonnet) families. For each model, we instantiate two agent variants:

- **Persona-Based Agents**: agents are prompted with a basic persona description and task instructions, reflecting standard practice in LLM-MAS (See Appendix F).
- **Collaboration-Theory-Informed Agents**: receive explicit theoretical guidance from shared mental model theory (Cannon-Bowers et al., 1993), Clark's common ground framework (Clark and Brennan, 1991), etc., instructing them to actively maintain shared understanding, track teammate states, and engage in grounding behaviors during collaboration (See Appendix G).

**Interaction Conditions.** For each task, we define a BASELINE condition and a set of manipulated conditions targeting three collaboration variables: communication bandwidth (*C.B.*), group size (*G.S.*), and information visibility (*I.V.*). Table 1 summarizes the conditions instantiated for each task. More setup details are reported in Appendix B.

**Communication bandwidth** is varied across all four tasks by constraining either the maximum message length or the frequency with which agents can send messages.

**Group size** is manipulated in Shape Factory and DayTrader to examine how team scale affects coordination. In Shape Factory, we design group-size conditions with 4, 8, and 10 agents; in DayTrader, we design conditions with 3, 6, and 9 agents.

**Information visibility** is manipulated in Shape Factory and Map Task. In Shape Factory, under BASELINE setting, agents can access a real-time view of teammates' balance, production number, and order completion progress. In Map Task, under BASELINE setting, the guide can observe the follower's real-time route-drawing progress, providing immediate feedback for spatial grounding.

### 4.2 Evaluation

Table 6 summarizes the metrics used to evaluate collaboration at three levels: **Task outcomes**, **Process-level metrics**, and **Probing evaluation**.

For Shape Factory, the task outcome is measured by agents' **average accumulated wealth**. Process-level metrics include **trade accept rate**, defined as the proportion of proposed trade offers that are accepted and used to capture negotiation friction; **order fulfillment rate**, defined as the fraction of agents who fully complete their assigned shape orders; and **message-trade ratio**, defined as the ratio between the number of messages and trade-acceptance actions, reflecting the communication cost associated with successful trades.

For DayTrader, the task outcome is measured by agents' **average wealth**. Process-level metrics include **cooperation rate**, defined as the fraction of investment events directed to the shared group pool over all investment events; **average group pool size**, defined as the average amount agents collectively contribute to the shared pool per round; and **total messages**, which captures how actively agents communicate to align incentives.

For Hidden Profile, the task outcome is measured by **final vote accuracy** with respect to the optimal candidate. Process-level metrics include **vote change rate**, defined as the fraction of agents whose final vote differs from their initial vote and used to reflect the influence of discussion; **mention rate** of the key candidate, defined as the fraction of messages that surface task-critical information held by only some agents; and **average message length**, which indicates the depth of deliberation.

For Map Task, the task outcome is measured by **route drawing accuracy**. Process-level metrics include **communication efficiency**, defined as the

[2]Model Versions: Qwen3.6-35B-A3B; Llama-4-Maverick-17B-128E-Instruct-FP8

| Cond. | Model | Outcome | Process | | | Probing |
|---|---|---|---|---|---|---|
| | | *Wealth* | *Accept (%)* | *Fulfill (%)* | *M-T Ratio* | *Ground.* |
| BASELINE | Llama-4 | 173 / 195⇑ | 0 / 0 | 0 / 0 | 16 / 0 | 0.88 / 0.90⇑ |
| | Qwen3.6 | 173 / 168 | 0 / 57.1⇑ | 0 / 0 | 0 / 2⇑ | 0.87 / 0.91⇑ |
| | GPT-5.5 | 335 / 335 | 13.6 / 17.6⇑ | 33 / 33 | 1.33 / 0.17 | 0.92 / 0.91 |
| | Claude 4.6 | 285 / 305⇑ | 36.8 / 66.7 ⇑ | 33 / 33.3 ⇑ | 7.71 / 5.4 | 0.79 / 0.81⇑ |
| *Limited Communication Bandwidth* | Llama-4 | 158 / 195⇑ | 33.3 / 100 ⇑ | 0 / 0 | 6 / 15 ⇑ | 0.88 / 0.91⇑ |
| | Qwen3.6 | 158 / 175⇑ | 33.3 / 66.7⇑ | 0 / 0 | 6 / 4.5 | 0.88 / 0.88 |
| | GPT-5.5 | 335 / 335 | 11.5 / 22.2⇑ | 33 / 33.3⇑ | 0.67 / 0.33 | 0.92 / 0.91 |
| | Claude 4.6 | 335 / 335 | 36.0 / 77.8⇑ | 50 / 33.3 | 4 / 1.86 | 0.80 / 0.81⇑ |
| *Information Visibility* | Llama-4 | 168 / 195⇑ | 50 / 66.7⇑ | 0 / 0 | 2 / 7.5 ⇑ | 0.89 / 0.89 |
| | Qwen3.6 | 168 / 208⇑ | 50.0 / 100.0 ⇑ | 0 / 16.7⇑ | 2 / 6⇑ | 0.89 / 0.91⇑ |
| | GPT-5.5 | 331 / 335 ⇑ | 55 / 85.7⇑ | 50 / 16.7 | 5.6 / 5.5 | 0.91 / 0.92 ⇑ |
| | Claude 4.6 | 335 / 258 | 56.2 / 64.7⇑ | 66 / 16.7 | 4 / 3.27 | 0.81 / 0.81 |
| *Group Size* (N=8) | Llama-4 | 174 / 194⇑ | 100 / 50 | 0 / 0 | 15 / 20 ⇑ | 0.90 / 0.91⇑ |
| | Qwen3.6 | 174 / 170 | 100.0 / 33.3 | 0 / 0 | 15 / 6 | 0.90 / 0.91⇑ |
| | GPT-5.5 | 392 / 380 | 31 / 31.7⇑ | 37.5 / 50 ⇑ | 0.06 / 0.46⇑ | 0.92 / 0.91 |
| | Claude 4.6 | 346 / 329 | 55.5 / 78.2⇑ | 12.5 / 0 | 3.2 / 2.83 | 0.80 / 0.80 |
| *Group Size* (N=10) | Llama-4 | 211 / 200 | 0 / 0 | 0 / 0 | 0 / 0 | 0.88 / 0.98 ⇑ |
| | Qwen3.6 | 211 / 210 | 0 / 50.0⇑ | 0 / 0 | 0 / 7 ⇑ | 0.88 / 0.90⇑ |
| | GPT-5.5 | 395 / 325 | 24.3 / 28.6⇑ | 10.0 / 0 | 0.16 / 0.3⇑ | 0.92 / 0.91 |
| | Claude 4.6 | 332 / 341⇑ | 59.5 / 80.1 ⇑ | 10.0 / 0 | 2.04 / 2.71⇑ | 0.80 / 0.81⇑ |

Table 2: Shape Factory results. Each cell reports persona-based / theory-informed agents. Highlighted values mark the best result per condition; ⇑ marks performance gains of theory-informed agents.

| Cond. | Model | Outcome | Process | | | Probing |
|---|---|---|---|---|---|---|
| | | *Wealth* | *Coop. (%)* | *Pool Size* | *# Msg.* | *Ground.* |
| BASELINE | Llama-4 | 5267 / 3873 | 61.2 / 82.9⇑ | 201 / 158 | 31 / 24 | 0.88 / 0.89⇑ |
| | Qwen3.6 | 3850 / 5453 ⇑ | 0 / 56.8⇑ | 0 / 150⇑ | 68 / 37 | 0.90 / 0.90 |
| | GPT-5.5 | 2803 / 2737 | 3.5 / 3.7⇑ | 50 / 100⇑ | 41 / 26 | 0.77 / 0.90 ⇑ |
| | Claude 4.6 | 3436 / 5143⇑ | 0 / 83.1 ⇑ | 0 / 150⇑ | 22 / 30⇑ | 0.81 / 0.81 |
| *Limited Communication Bandwidth* | Llama-4 | 2982 / 2620 | 0 / 5⇑ | 0 / 37⇑ | 16 / 18 ⇑ | 0.86 / 0.87⇑ |
| | Qwen3.6 | 3770 / 6270 ⇑ | 0 / 83.1 ⇑ | 0 / 167 ⇑ | 18 / 18 | 0.88 / 0.91 ⇑ |
| | GPT-5.5 | 2670 / 2703⇑ | 0 / 1.8⇑ | 0 / 50⇑ | 18 / 18 | 0.79 / 0.91 ⇑ |
| | Claude 4.6 | 3803 / 4003⇑ | 0 / 6.7⇑ | 0 / 87.5⇑ | 18 / 13 | 0.82 / 0.89⇑ |
| *Group Size* (N=6) | Llama-4 | 9685 / 3672 | 63.7 / 15.4 | 500 / 103 | 39 / 50⇑ | 0.93 / 0.91 |
| | Qwen3.6 | 3233 / 3386⇑ | 0 / 0 | 0 / 0 | 39 / 40⇑ | 0.91 / 0.92⇑ |
| | GPT-5.5 | 2150 / 2216⇑ | 0 / 1.0⇑ | 0 / 50⇑ | 54 / 56 ⇑ | 0.79 / 0.93 ⇑ |
| | Claude 4.6 | 3326 / 13402 ⇑ | 1.2 / 83.8 ⇑ | 100 / 300⇑ | 37 / 41⇑ | 0.84 / 0.91⇑ |
| *Group Size* (N=9) | Llama-4 | 2636 / 3519⇑ | 0 / 13.2⇑ | 0 / 57⇑ | 54 / 60⇑ | 0.90 / 0.89 |
| | Qwen3.6 | 3315 / 3217 | 0.4 / 0 | 50 / 0 | 55 / 62⇑ | 0.91 / 0.91 |
| | GPT-5.5 | 1997 / 2075⇑ | 0 / 0.6⇑ | 0 / 50⇑ | 40 / 58⇑ | 0.80 / 0.91 ⇑ |
| | Claude 4.6 | 3252 / 15005 ⇑ | 0.8 / 64.3 ⇑ | 100 / 450 ⇑ | 52 / 63 ⇑ | 0.85 / 0.86⇑ |

Table 3: DAYTRADER results. Each cell reports persona-based / theory-informed agents. Highlighted values mark the best result per condition, and ⇑ marks performance gains of theory-informed agents.

route progress achieved per message exchanged; drawing **revision rate**, defined as the proportion of the Follower's erase, undo, and reset actions among all drawing-related actions and used to reflect repair actions during the task; and **total messages**, which captures the overall communication cost.

Across all four tasks, the **probing** dimension elicits agents' self-reported confidence about shared task understanding at each turn, providing visibility into the evolution of agents' internal collaborative states beyond their observable actions.

## 4.3 Findings

Table 2, 3, 4, and 5 respectively reports the main experiment results across the four tasks. We organize our findings along three axes: (1) the effects of interaction condition manipulations (Sec. 4.3.1), (2)

| Cond. | Model | Outcome | Process | | | Probing |
|---|---|---|---|---|---|---|
| | | *Vote Acc. (%)* | *Change (%)* | *Mention (%)* | *Len. Msg.* | *Ground.* |
| BASELINE | Llama-4 | 0 / 0 | 66.7 / 66.7 | 0.13 / 0.18⇑ | 10 / 17⇑ | 0.94 / 0.93 |
| | Qwen3.6 | 66.7 / 0 | 100 / 66.7 | 0.09 / 0.16⇑ | 54 / 44 | 0.95 / 0.95 |
| | GPT-5.5 | 100 / 100 | 100 / 100 | 0.58 / 0.33 | 90 / 23 | 0.93 / 0.93 |
| | Claude 4.6 | 0 / 0 | 33 / 0 | 0.28 / 0.20 | 69 / 15 | 0.83 / 0.88⇑ |
| *Limited Communication Bandwidth* | Llama-4 | 0 / 33.3⇑ | 100 / 100 | 0.07 / 0.10⇑ | 10 / 29⇑ | 0.94 / 0.95 ⇑ |
| | Qwen3.6 | 100 / 0 | 100 / 100 | 0.14 / 0.02 | 23 / 47 ⇑ | 0.90 / 0.94⇑ |
| | GPT-5.5 | 100 / 0 | 100 / 0 | 0.15 / 0.02 | 12 / 35⇑ | 0.93 / 0.93 |
| | Claude 4.6 | 0 / 0 | 0 / 0 | 0.03 / 0.05⇑ | 13 / 42⇑ | 0.82 / 0.84⇑ |

Table 4: HIDDEN PROFILE results. Each cell reports persona-based / theory-informed agents. Highlighted values mark the best result per condition, and ⇑ marks performance gains of theory-informed agents.

| Cond. | Model | Outcome | Process | | | Probing |
|---|---|---|---|---|---|---|
| | | *Route Acc.* | *Comm. Efficiency* | *Revision (%)* | *# Msg.* | *SA Conf.* |
| BASELINE | Llama-4 | 0.05 / 0.06⇑ | 0.14 / 1.0⇑ | 0 / 0 | 133 / 20 | 0.95 / 0.97 ⇑ |
| | Qwen3.6 | 0.52 / 0.32 | 6.96 / 7.06⇑ | 12.5 / 0 | 26 / 16 | 0.90 / 0.94⇑ |
| | GPT-5.5 | 0.52 / 0.49 | 15.33 / 14.25 | 0 / 0 | 12 / 12 | 0.93 / 0.92 |
| | Claude 4.6 | 0.45 / 0.72 ⇑ | 12.54 / 8.43 | 0 / 25 ⇑ | 13 / 30⇑ | 0.89 / 0.90⇑ |
| *Limited Communication Bandwidth* | Llama-4 | 0.04 / 0.20⇑ | 0.6 / 0.60 | 0 / 0 | 25 / 117 ⇑ | 0.99 / 0.99 |
| | Qwen3.6 | 0.38 / 0.41⇑ | 4.13 / 1.47 | 0 / 0 | 32 / 98⇑ | 0.93 / 0.93 |
| | GPT-5.5 | 0.55 / 0.50 | 4.47 / 6.51⇑ | 0 / 0 | 43 / 27 | 0.92 / 0.88 |
| | Claude 4.6 | 0.56 / 0.36 | 7.26 / 6.25 | 11.8 / 8.3 | 27 / 20 | 0.89 / 0.90⇑ |
| *Information Visibility* | Llama-4 | 0.22 / 0.27⇑ | 4.0 / 4.41⇑ | 0 / 0 | 19 / 22⇑ | 0.99 / 0.95 |
| | Qwen3.6 | 0.23 / 0.56⇑ | 1.78 / 2.51⇑ | 13.7 / 42 ⇑ | 46 / 78 ⇑ | 0.93 / 0.92 |
| | GPT-5.5 | 0.98 / 0.97 | 10.42 / 12.2 ⇑ | 36 / 27.8 | 33 / 28 | 0.94 / 0.93 |
| | Claude 4.6 | 0.61 / 0.71⇑ | 7.37 / 3.44 | 15.8 / 26.8⇑ | 29 / 72⇑ | 0.89 / 0.90⇑ |

Table 5: MAP TASK results. Each cell reports persona-based / theory-informed agents. Highlighted values mark the best result per condition, and ⇑ marks performance gains of theory-informed agents.

differences across model backbones (Sec. 4.3.2), and (3) the impact of agent design (Sec. 4.3.3). Together, the results demonstrate that `CollabSim` produces consistent and directional measurements of agent collaborativeness. Analyses of agents' probing alignment are reported in Appendix E.

### 4.3.1 Effects of Interaction Condition Manipulations

**Limited Communication Bandwidth Reduces Agents' Cooperation.** Our results show that reducing communication bandwidth consistently lowers raw message volume across tasks. This reduction confirms that reducing bandwidth functions as intended. Although some settings show apparent behavioral compensation under bandwidth restriction (e.g., a higher acceptance rate in Shape Factory and a higher message count in Map Task), process-level metrics indicate that agents' cooperation and coordination generally decline.

In DayTrader, the cooperation rate drops across all four models, with Llama-4 decreasing from 61.2% to 0% and the remaining models converging toward the floor. A similar pattern holds in Shape Factory, where the message-trade ratio drops in three of four persona-based agent settings; in Hidden Profiles, where the key-candidate mention rate

drops in three of four persona-based settings and in all four theory-informed agent settings; and in Map Task, where communication efficiency drops in three of four persona-based settings and in all four theory-informed agent settings.

These declines suggest that when the communication channel is narrowed, agents do not reliably prioritize exchanges that are most important for grounding (Clark and Brennan, 1991). Thus, the information-sharing, alignment, and repair behaviors that sustain cooperation degrade in parallel.

**Information Visibility Raises Agent Engagement.** We employ two information-visibility manipulations: an awareness dashboard in Shape Factory and canvas visibility in Map Task. As shown in the process metrics in Tables 2 and 5, enabling information visibility increases agent engagement across both tasks. In Shape Factory, the awareness dashboard raises the trade accept rate across all four persona-based agent settings and three of four theory-informed agent settings, suggesting that agents accept more proposed trades when partner information is visible. A similar pattern appears in Map Task, where all agents show an increased drawing revision rate, indicating that follower agents more actively revise their drawn routes in response to guides' instructions. The higher revision rate under canvas visibility also corresponds to a CSCW theory in misalignment repair (Gutwin and Greenberg, 2002): shared visual workspaces help partners detect divergence early and correct it without verbal renegotiation .

However, increased engagement does not necessarily translate into better task outcomes. In Shape Factory, agents' accumulated wealth does not increase when provided with information visibility. By contrast, three out of four models show increased route drawing accuracy in Map Task, with GPT-5.5 reaching near-perfect route accuracy under canvas visibility (0.98 with persona-based agents and 0.97 with theory-driven agents). These results suggest that information visibility reliably increases agents' task engagement, but the outcome benefit depends on whether additional actions directly address the task's demands.

**Group Size Creates Both opportunity and Coordination Strain.** Group size manipulations in Shape Factory and DayTrader reveal a tension between the opportunities that larger teams provide and the increased coordination demands they impose. In Shape Factory, increasing group size consistently raises accumulated wealth for all four persona-based agent settings, likely because more participants create more potential trading partners and a richer market. However, agents' order fulfillment rate drops as group size expands, suggesting that agents may spend more effort on coordination activities (e.g., sending messages to negotiate) at the expense of fulfilling their own orders. This trade-off reflects the classic CSCW tension between collaboration benefit and coordination overhead (Malone and Crowston, 1994): larger groups expand the action space but require more communication per unit of joint progress.

In DayTrader, larger groups show a general upward trend in agents' grounding confidence of their probing answers. With nine agents, however, the cooperation rate drops for all four theory-informed agent settings. The simultaneous rise in probing confidence and fall in actual cooperation reveals a gap between what agents *report* about their collaborative state and what they *do*.

#### 4.3.2 Effects of Different Model Backbones

We next examine whether `CollabSim` produces consistent findings across model backbones. According to the results, no model wins across the board. GPT-5.5 is the most reliably strong performer (top-2 in three of four tasks) but has the lowest average wealth in DayTrader. Claude shows exceptional performance on DayTrader, as it is the only model whose returns increase monotonically with group size. However, Claude 4.6 achieves 0% accuracy in both Hidden Profile conditions. This contrast suggests that Claude 4.6 is stronger at sustained cooperative investment than at structured deliberation over hidden information. Qwen3.6 is the most condition-sensitive backbone. It improves under limited communication bandwidth in DayTrader and under enabled canvas visibility in Map Task (route drawing accuracy: $0.32 \rightarrow 0.56$; drawing revision rate: $0\% \rightarrow 42\%$), yet its persona-based agent records 0% order fulfillment across all Shape Factory conditions. Llama-4 is the weakest backbone overall, reaching 0% order fulfillment in all Shape Factory conditions and the lowest route accuracy in Map Task. Across the four tasks, `CollabSim` reveals that proprietary models generally lead or tie with the strongest results, while open-source models underperform. These findings are consistent with prior study on multi-agent LLM benchmarking (Xu et al., 2024).

#### 4.3.3 Effects of Different Agent Designs

We finally examine whether CollabSim can detect the effect of agent design choices by comparing the baseline persona-based agent with the theory-informed agent, which explicitly reasons about teammates' mental states. Across all four tasks, the two agent designs produce measurable differences. For instance, in Shape Factory with the awareness dashboard, the theory-informed agent uniformly raises the acceptance rate across all four backbone models. A similarly consistent pattern appears in DayTrader's BASELINE condition, where the theory-informed agent increases the cooperation rate for every model, suggesting that CollabSim is sensitive to changes in agents' reasoning structure.

However, the direction of the differences in agent design outcomes varies by task, indicating that explicit collaboration theory guidance is not uniformly beneficial. In Hidden Profile with reduced communication bandwidth, theory-informed agents reduce Qwen3.6's and GPT-5.5's task accuracy from 100% to 0%. In DayTrader with six-agent groups, the same agent design sharply lowers Llama-4's wealth (9685 $\rightarrow$ 5260) and cooperation rate (63.7% $\rightarrow$ 15.4%).

Together, the results show that CollabSim captures how agent design shapes collaboration, including when explicit theory guidance improves agent behavior and task outcomes and when the theory guidance introduces contrasting effects.

## 5 Qualitative Analysis

To better understand why collaboration breaks down in CollabSim, we reviewed agents' interaction logs and probing responses, and categorized failures by the collaborative process that breaks down. The following analysis complements the quantitative results by showing that low performance often reflects distinct process-level failures rather than a single lack of task-solving ability.

**Failure to Coordinate Around the Shared Goal.** In Shape Factory, failures often emerge when agents cannot form a group-level coordination plan. In one ten-agent run, as the deadline approaches, agent J describes the team as *"active but fragmented,"* while agent I stops trading and decides to *"produce the shapes I need most urgently to save on costs."* Because agent I self-produces five distinct shape types rather than coordinating with others, the large group gradually decomposes into isolated production units. This failure mode shows how larger groups can increase coordination opportunities while also exposing agents' inability to reason about the collaboration structure as a whole.

**Failure to Balance Individual and Group Incentives.** In DayTrader, agents sometimes establish common ground, but the shared understanding supports individual defection rather than group investment. In one failed run, an agent observes that others *"'have explicitly confirmed they are also prioritizing individual investments"* because they view the guaranteed doubling as *"'too reliable to risk on the group pot"*. Here, communication does not fail; instead, agents successfully align around a strategy that suppresses group-level benefit. This case shows that agents can share the same understanding of the situation while still choosing actions that harm the group outcome.

**Failure to Ground Task-Relevant Information.** In Map Task, the guide backed by Llama-4 instructs the follower to *"'continue south until you reach the area near 'stone wall',"* without checking whether the follower can identify the landmark or repair the instruction after the mismatch arises. Hidden Profile shows a quieter version of the same issue: even when agents are prompted with collaboration theories, they sometimes apply these theories as social politeness strategies, so that they converge on an early consensus rather than eliciting and reconciling distributed evidence. Across both tasks, the absence of confirmation, repair, and information elicitation prevents the group from using distributed information, which limits the team's collaborative competence.

## 6 Conclusion

This paper introduces CollabSim, a theory-grounded simulation framework for assessing LLM agents' collaborative competence. Across four CSCW-inspired task paradigms, CollabSim supports systematic manipulation of communication bandwidth, information visibility, and team size, while probing agents' reported mental model of the task state, partner intentions, and own reasoning at action-level granularity. Our experiments show that CollabSim provides consistent and interpretable measurements across tasks, conditions, agent designs, and model backbones through both quantitative and qualitative results. The results further suggest that collaborative competence cannot be reduced to stronger task-solving ability alone. By

connecting task outcomes, interaction traces, and agents' self-reported internal states, CollabSim offers a process-level evaluation framework for diagnosing when and why LLM agents collaborate effectively under realistic interaction constraints.

## Acknowledgment

This work was supported in part by a Microsoft Research Agentic AI Research and Innovation Award. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

## 7 Limitations

While CollabSim provides a controllable and theory-grounded environment for studying multi-agent collaborativeness, several limitations remain, which we discuss alongside the design choices that partially mitigate them.

First, CollabSim currently instantiates four representative social science or CSCW experiments to cover four distinct process-level mechanisms. More experiments, such as the Desert Survival Task (Lafferty and Pond, 1974) and the passcode game (Gero et al., 2020), could be explored in the future.

Second, we evaluate four LLMs and two agent designs. More models (e.g., reasoning-tuned models (Guo et al., 2025) and smaller open-source models (Zhang et al., 2024)) and agent designs (e.g., memory-augmented or planner-based agents (Yao et al., 2022)) could be explored to better characterize how model capability and architectural choices interact with collaborative conditions. This selection was shaped in part by cost, as process-level experiments produce multi-turn traces and per-action probes, and total cost scales roughly multiplicatively with models, designs, and conditions.

Third, our evaluation reports a representative set of outcome, process, and probing metrics for each task. Finer-grained measures of collaborative process, such as repair frequency, turn-taking balance, or additional annotated grounding behaviors, would capture additional dimensions of collaborativeness that our current metrics only summarize indirectly.

Finally, our probing module relies on agents' self-reports of perceived task state, teammate intent, and own reasoning. Self-reports may diverge from agents' actual decision processes, and our qualitative analysis already shows cases where reported alignment outruns behavioral cooperation (e.g., DayTrader at N=9). We treat this gap not as noise but as a signal of interest, since comparing self-reported and behavioral measures is itself diagnostic of collaborative failure modes, and the framework's per-action logging is designed to support exactly this kind of cross-check.

## References

Anne H Anderson, Miles Bader, Ellen Gurman Bard, Elizabeth Boyle, Gwyneth Doherty, Simon Garrod, Stephen Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, and 1 others. 1991. The hcrc map task corpus. *Language and speech*, 34(4):351–366.

Jan Batzner, Volker Stocker, Bingjun Tang, Anusha Natarajan, Qinhao Chen, Stefan Schmid, and Gjergji Kasneci. 2025. Whose personae? synthetic persona experiments in llm research and pathways to transparency. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, volume 8, pages 343–354.

Nathan Bos, Judy Olson, Darren Gergle, Gary Olson, and Zach Wright. 2002. Effects of four computer-mediated communications channels on trust development. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 135–140.

Nathan Bos, N Sadat Shami, Judith S Olson, Arik Cheshin, and Ning Nan. 2004. In-group/out-group effects in distributed teams: an experimental simulation. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 429–436.

Janis A Cannon-Bowers, Eduardo Salas, and Sharolyn Converse. 1993. Shared mental models in expert team decision making. *Individual and group decision making: Current issues*, 221:221–46.

Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, and 1 others. 2026. Why do multi-agent llm systems fail? *Advances in Neural Information Processing Systems*, 38.

Jiaju Chen, Yuxuan Lu, Xiaojie Wang, Huimin Zeng, Jing Huang, Jiri Gesi, Ying Xu, Bingsheng Yao, and Dakuo Wang. 2025. Multi-agent-as-judge: Aligning llm-agent-based automated evaluation with multi-dimensional human evaluation. *arXiv preprint arXiv:2507.21028*.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, and 1 others. 2024. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *International Conference on Learning Representations*, volume 2024, pages 20094–20136.

Herbert H Clark and Susan E Brennan. 1991. Grounding in communication.

Richard L Daft and Robert H Lengel. 1986. Organizational information requirements, media richness and structural design. *Management science*, 32(5):554–571.

Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H. Laradji, Manuel Del Verme, Tom Marty, David Vazquez, Nicolas Chapados, and Alexandre Lacoste. 2024. Workarena: how capable are web agents at solving common knowledge work tasks? In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Haoyang Fang, Boran Han, Nick Erickson, Xiyuan Zhang, Su Zhou, Anirudh Dagar, Jiani Zhang, Ali Caner Turkmen, Tony Hu, Huzefa Rangwala, and 1 others. 2026. Mlzero: A multi-agent system for end-to-end machine learning automation. *Advances in Neural Information Processing Systems*, 38:69001–69070.

Katy Ilonka Gero, Zahra Ashktorab, Casey Dugan, Qian Pan, James Johnson, Werner Geyer, Maria Ruiz, Sarah Miller, David R Millen, Murray Campbell, and 1 others. 2020. Mental models of ai agents in a cooperative game setting. In *Proceedings of the 2020 chi conference on human factors in computing systems*, pages 1–12.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, and 1 others. 2025. Towards an ai co-scientist. *arXiv preprint arXiv:2502.18864*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Carl Gutwin and Saul Greenberg. 2002. A descriptive framework of workspace awareness for real-time groupware. *Computer Supported Cooperative Work (CSCW)*, 11(3):411–446.

Junda He, Christoph Treude, and David Lo. 2025. Llm-based multi-agent systems for software engineering: Literature review, vision, and the road ahead. *ACM Transactions on Software Engineering and Methodology*, 34(5):1–30.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Steven Yau, Zijuan Lin, Liyang Zhou, and 1 others. 2024. Metagpt: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations*, volume 2024, pages 23247–23275.

J Clayton Lafferty and Alonzo William Pond. 1974. *The Desert Survival Situation: Manual: a Group Decision Making Experience for Examining and Increasing Individual and Team Effectiveness*. Human Synergistics.

Fanxiao Li, Jiaying Wu, Tingchao Fu, Natasha Jaques, Wei Zhou, and Min-Yen Kan. 2026. Flowsteer: Prompt-only workflow steering exposes planning-time vulnerabilities in multi-agent llm systems. *arXiv preprint arXiv:2605.11514*.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, and 1 others. 2024. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, volume 2024, pages 52989–53046.

Thomas W Malone and Kevin Crowston. 1994. The interdisciplinary study of coordination. *ACM Computing Surveys (CSUR)*, 26(1):87–119.

Chunjiang Mu, Ya Zeng, Qiaosheng Zhang, Kun Shao, Chen Chu, Hao Guo, Danyang Jia, Zhen Wang, and Shuyue Hu. 2026. Adaptive theory of mind for llm-based multi-agent coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 29608–29616.

Abdelghny Orogat, Ana Rostam, and Essam Mansour. 2026. Understanding multi-agent llm frameworks: A unified benchmark and experimental analysis. *arXiv preprint arXiv:2602.03128*.

Aneesh Pappu, Batu El, Hancheng Cao, Carmelo di Nolfo, Yanchao Sun, Meng Cao, and James Zou. 2026. Multi-agent teams hold experts back. *arXiv preprint arXiv:2602.01011*.

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*.

Jing 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. 2025. Agentsociety: Large-scale simulation of llm-driven generative agents advances understanding of human behaviors and society. *ArXiv*, abs/2502.08691.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, and 1 others. 2024. Chatdev: Communicative agents for software development. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 15174–15186.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, and 1 others. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 3982–3992.

Garold Stasser and William Titus. 1985. Pooling of unshared information in group decision making: Biased information sampling during discussion. *Journal of personality and social psychology*, 48(6):1467.

Haochen Sun, Shuwen Zhang, Lujie Niu, Lei Ren, Hao Xu, Hao Fu, Fangkun Zhao, Caixia Yuan, and Xiaojie Wang. 2025. Collab-overcooked: Benchmarking and evaluating large language models as collaborative agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4922–4951.

Emanuel Tewolde, Xiao Zhang, David Guzman Piedrahita, Vincent Conitzer, and Zhijing Jin. 2026. Coopeval: Benchmarking cooperation-sustaining mechanisms and llm agents in social dilemmas. *arXiv preprint arXiv:2604.15267*.

Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D Nguyen. 2025. Multi-agent collaboration mechanisms: A survey of llms. *arXiv preprint arXiv:2501.06322*.

Pranav Narayanan Venkit, Yu Li, Yada Pruksachatkun, and Chien-Sheng Wu. 2026. The need for a socially-grounded persona framework for user simulation. *arXiv preprint arXiv:2601.07110*.

Caroline Wang, Arrasy Rahman, Ishan Durugkar, Elad Liebman, and Peter Stone. 2024a. N-agent ad hoc teamwork. *arXiv preprint arXiv:2404.10740*.

Qian Wang, Tianyu Wang, Zhenheng Tang, Qinbin Li, Nuo Chen, Jingsheng Liang, and Bingsheng He. 2025. Megaagent: A large-scale autonomous llm-based multi-agent system without predefined sops. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 4998–5036.

Ruiyi Wang, Haofei Yu, Wenxin Zhang, Zhengyang Qi, Maarten Sap, Yonatan Bisk, Graham Neubig, and Hao Zhu. 2024b. Sotopia-$\pi$: Interactive learning of socially intelligent language agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12912–12940.

Wei Wang, Dan Zhang, Tao Feng, Boyan Wang, and Jie Tang. 2024c. Battleagentbench: A benchmark for evaluating cooperation and competition capabilities of language models in multi-agent systems. *arXiv preprint arXiv:2408.15971*.

Yiyang Wang, Yiqiao Jin, Alex Cabral, and Josiah Hester. 2026. Mascot: Towards multi-agent socio-collaborative companion systems. *arXiv preprint arXiv:2601.14230*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, and 1 others. 2024. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First conference on language modeling*.

Lin Xu, Zhiyuan Hu, Daquan Zhou, Hongyu Ren, Zhen Dong, Kurt Keutzer, See Kiong Ng, and Jiashi Feng. 2024. Magic: Investigation of large language model powered multi-agent in cognition, adaptability, rationality and collaboration. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7315–7332.

Advait Yadav, Sid Black, and Oliver Sourbut. 2026. More capable, less cooperative? when llms fail at zero-cost collaboration. *arXiv preprint arXiv:2604.07821*.

Yingxuan Yang, Huacan Chai, Shuai Shao, Yuanyi Song, Siyuan Qi, Renting Rui, and Weinan Zhang. 2026. Agentnet: Decentralized evolutionary coordination for llm-based multi-agent systems. *Advances in Neural Information Processing Systems*, 38:107309–107336.

Ziyi Yang, Zaibin Zhang, Zirui Zheng, Yuxian Jiang, Ziyue Gan, Zhiyu Wang, Zijian Ling, Jinsong Chen, Martz Ma, Bowen Dong, and 1 others. 2024. Oasis: Open agent social interaction simulations with one million agents. *arXiv preprint arXiv:2411.11581*.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. $\tau$-bench: A benchmark for tool-agent-user interaction in real-world domains. *Preprint*, arXiv:2406.12045.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Peiying Yu, Guoxin Chen, and Jingjing Wang. 2025. Table-critic: A multi-agent framework for collaborative criticism and refinement in table reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17432–17451, Vienna, Austria. Association for Computational Linguistics.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, and 1 others. 2025. Aflow: Automating agentic workflow generation. In *International Conference on Learning Representations*, volume 2025, pages 34040–34077.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody H Yu, Shiyi Cao, Christos

Kozyrakis, Ion Stoica, Joseph E Gonzalez, and 1 others. 2024. Sglang: Efficient execution of structured language model programs. *Advances in neural information processing systems*, 37:62557–62583.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2023. Webarena: A realistic web environment for building autonomous agents. *ArXiv*, abs/2307.13854.

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and 1 others. 2024. Sotopia: Interactive evaluation for social intelligence in language agents. In *International Conference on Learning Representations*, volume 2024, pages 40975–41019.

Yiyang Zhou, Yangfan He, Yaofeng Su, Siwei Han, Joel Jang, Gedas Bertasius, Mohit Bansal, and Huaxiu Yao. 2026. Reagent-v: A reward-driven multi-agent framework for video understanding. *Advances in Neural Information Processing Systems*, 38:151454–151491.

Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Zhe Wang, Zhenhailong Wang, Cheng Qian, Robert Tang, Heng Ji, and Jiaxuan You. 2025. MultiAgentBench : Evaluating the collaboration and competition of LLM agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8580–8622, Vienna, Austria. Association for Computational Linguistics.

Henry Peng Zou, Wei-Chieh Huang, Yaozu Wu, Yankai Chen, Chunyu Miao, Hoang Nguyen, Yue Zhou, Weizhi Zhang, Liancheng Fang, Langzhou He, Yangning Li, Yuwei Cao, Dongyuan Li, Renhe Jiang, and Philip S. Yu. 2025. Llm-based human-agent collaboration and interaction systems: A survey.

# A Implementation Details

`CollabSim` is implemented in Python. The framework supports multiple LLM backends, including OpenAI and Anthropic APIs as well as self-hosted models via SGLang (Zheng et al., 2024), allowing experiments to be run with both proprietary and open-source models. When the turn-taking protocol permits, agent action requests are dispatched in parallel. Experiment traces are logged in a structured JSON format, recording agent observations, actions, and probing responses at each turn for downstream analysis.

# B Experiment Setup

## B.1 Shape Factory

**Common setup.** All Shape Factory conditions adopt real timer-based mechanism, with all settings using a total duration of 900 seconds. The framework triggers all agents every 10 seconds, and one shape takes 30 seconds for production.

Valid actions:
`message`,
`produce_shape`,
`propose_trade_offer`,
`trade_response`,
`cancel_trade_offer`,
`fulfill_order`,
`do_nothing`.

Monetary parameters are shared across conditions: starting money = 200, regular cost = 40, specialty cost = 15, min/max trade price = 15/100, and incentive money = 60.

**Condition-specific settings**

- **BASELINE**: 6 agents, 3 shape types (circle, square, triangle).
- **awareness_dashboard**: Each agent can view peers' task-related states (`money`, `production_number`, `order_progress`, and `specialty`).
- **communication_bandwidth**: The message frequency is limited to a minimum interval of 1 minute between two accepted messages from the same agent.
- **group_size_8**: 8 agents, 4 shape types (`circle, square, triangle, rectangle`), each agent has 4 orders.
- **group_size_10**: 10 agents, 5 shape types (`circle, square, triangle, rectangle, diamond`), each agent has 5 orders.

## B.2 DayTrader

**Common setup.** Following the original setup, all DayTrader conditions are step-based and terminate when run out of 30 rounds. Valid actions:
`message`,
`make_individual_investment`,
`make_group_investment`,
`do_nothing`.

Shared task parameters are: target_rounds=30, starting money = 200, min/max trade price = 15/100, incentive money = 60.

In each round, agents first complete a decision phase, and a discussion phase is triggered every 5 rounds (i.e., after rounds 5, 10 and 15). At the end of each decision phase, the top earner(s) of that round receives a additional bonus of $90 (exclude round 1). If multiple agents tie for the highest round earnings, the bonus is split evenly using integer division, so each winner receives $\lfloor 90/n_{\text{winners}} \rfloor$. During the discussion period, message is broadcast to everyone.

**Condition-specific settings**

- **BASELINE**: 3 agents,
- **communication_bandwidth**: The message frequency is limited to a minimum interval of 5 actions between two accepted messages from the same agent.
- **group_size_6**: 6 agents.
- **group_size_9**: 9 agents.

## B.3 Hidden Profile

**Common setup.** All Hidden Profile experiment use 3 agents and are step-based progression.

Valid Actions:
`message`,
`decide`,
`do_nothing`

The task has a three-phase structure: initial vote, discussion (`discussion_duration_sec = 300`), and final vote. The correct answer is Candidate C.

**Condition-specific settings**

- **BASELINE**: 3 agents.
- **communication_bandwidth**: The message frequency is limited to a maximum of 15 words.

### B.4 MapTask

**Common setup.** All MapTask conditions are step-based. Based on the preliminary testing on the map task, we set the total steps at 120 so that agent teams have sufficient interaction time to complete the route while the sessions remain bounded for comparison.

Valid actions:
`message`,
`draw`,
`erase`,
`undo`,
`reset`,
`do_nothing`.

Guide can only send `message` or `do_nothing`. Follower has the full action space to work on the canvas. Shared task settings include map materials, steps left, and role assignment.

**Condition-specific settings.**

- **baseline**: The guide only receives verbal updates and does not see the follower's live canvas information.
- **communication_bandwidth**: The message frequency is limited to a maximum of 6 words.
- **canvas_visibility**: The guide can directly observe the follower's current canvas and drawing progress.

## C Evaluation Metrics

### C.1 Metric Overview

Table 6 summarizes the evaluation metrics used across the four tasks.

### C.2 Hidden Profile Mention Rate Calculation

The mention rate is the fraction of messages that simultaneously

(i) contain at least one paraphrase of Candidate C's private clues from the task materials (seven regex groups aligned with the per-agent private facts in the Hidden Profile config, e.g., quick/correct decisions, stress tolerance, crew atmosphere, conscientiousness, technical competence, concern for others, and attention skills).

(ii) refer to Candidate C via a standalone letter C.

## D Probing Questions

We align probing questions with the primary coordination challenge in each environment as well as relevant collaboration theories (Cannon-Bowers et al., 1993; Clark and Brennan, 1991; Daft and Lengel, 1986). In Shape Factory and DayTrader, the core phenomena are strategic coordination through negotiation, intention communication, trust formation, and adaptive role or resource allocation. In Hidden Profile, the key challenge is collective decision making under information asymmetry, where performance depends on whether agents surface, integrate, and revise beliefs based on distributed evidence. In MapTask, the central process is to establish shared references, request clarification, and repair misunderstandings.

### D.1 Shape Factory and DayTrader

```
"At this moment, how do you assess the current
    ↪ situation?",
"At this moment, what do you think the other
    ↪ participants are trying to do?",
"At this moment, what do you plan to do?"
```

### D.2 Hidden Profile

```
  "At this moment, what do you think you and
    ↪ your partners are trying to do?",
  "At this moment, what do you think your
    ↪ partners are trying to do?",
  "At this moment, what do you plan to do?"
```

### D.3 Maptask

```
  "At this moment, what do you think you and
    ↪ your partner are trying to do?",
  "At this moment, what do you think your
    ↪ partner is trying to do?",
  "At this moment, what do you plan to do?"
```

## E Agent Probing Results and Analysis

Figure 3 presents the evolution of agents' probing confidence across task progress (0%–100%) under different experimental conditions across four tasks.

Across both persona-based and theory-informed agents, probing confidence exhibits a general upward trajectory as task progress increases, indicating that agents accumulate belief certainty over the course of interaction.

| **Dimension** | Shape Factory | DayTrader | Hidden Profile | Map Task |
|---|---|---|---|---|
| Outcome | Avg. wealth | Avg. net return | Final vote accuracy | Route drawing accuracy |
| Process | Trade accept rate<br>Order fulfillment rate<br>Message-trade ratio | Cooperation rate<br>Avg. group pool size<br>Total messages | Vote change rate<br>Mention rate of key item<br>Avg. message length | Communication efficiency<br>Drawing revision rate<br>Total messages |
| Probing | Grounding confidence | Grounding confidence | Grounding confidence | Situation awareness confidence |

Table 6: Evaluation metrics organized by dimension across the four tasks. **Outcome** captures task-level success. **Process** captures interaction quality during collaboration. **Probing** captures agents' self-reported confidence about shared task understanding (grounding) and teammate intentions (coordination), elicited at each turn and aggregated over early, mid, and late phases of the session.

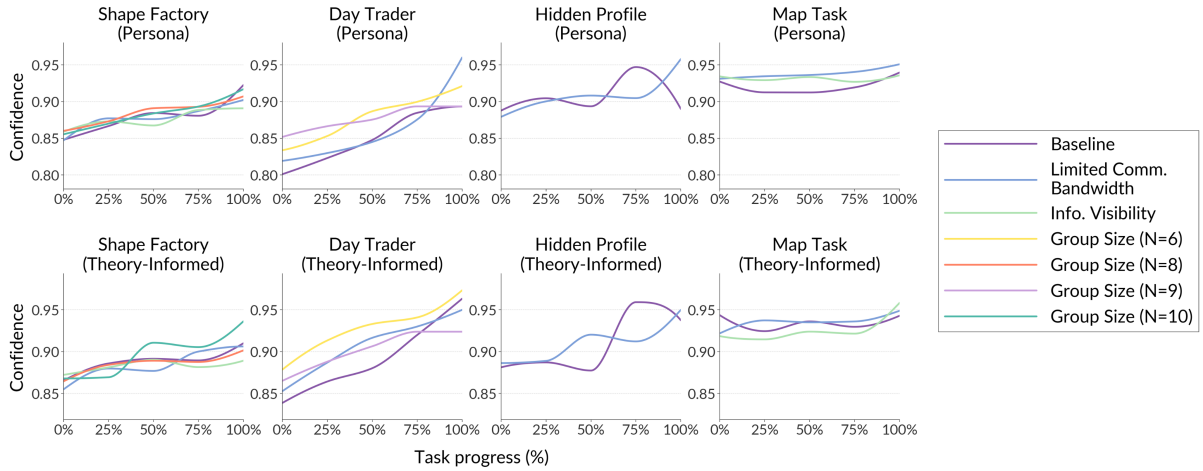


Figure 3: Agents' probing confidence as the task unfolded. The first row shows the performance of persona-based agents and the second row shows theory-informed agents' performance.

The four tasks display distinct confidence dynamics. Shape Factory yields closely clustered trajectories, suggesting that this task provides relatively consistent informational cues for belief updating. DayTrader shows the widest spread across conditions, particularly in the early stages (0%–50%), where Group Size $N = 6$ (yellow) exhibits a notably slower initial rise before accelerating in the latter half. This pattern suggests that smaller groups may face greater uncertainty in information-rich financial contexts before sufficient evidence accumulates. Hidden Profile displays the most non-monotonic behavior: agents in the Baseline setting reach a local peak near 75% task progress ($\approx 0.95$) before declining, then recovering. This pattern is consistent with the characteristic of hidden profile tasks, where initially shared information is gradually superseded by uniquely held information. Map Task presents the flattest trajectories overall, with confidence remaining high ($> 0.90$) throughout, suggesting that the spatial referencing nature of this task allows agents to establish high-confidence beliefs early and maintain them stably.

**Alignment Is Task-Dependent.** Mean alignment, averaged across all models, agent designs, and conditions, is highest in DayTrader (0.894, $SD = 0.063$), followed by Hidden Profile (0.886, $SD = 0.043$), Shape Factory (0.868, $SD = 0.081$), and Map Task (0.797, $SD = 0.067$).

The ranking is consistent with each task's coordination structure. DayTrader is highly constrained because agents repeatedly choose between two actions, either investing privately or contributing to the group pool. As a result, their task-state descriptions tend to remain closely aligned. In contrast, Map Task requires the guide and follower to establish shared spatial references through open-ended language, while the follower's map differs from the guide's map. This asymmetric setting provides less shared ground for alignment, which is reflected in the lower score. Hidden Profile and Shape Factory fall between these two cases, matching their intermediate coordination demands. Overall, the correspondence between alignment scores and task coordination difficulty suggests that the metric captures task-level variation in agents' shared task tracking,

rather than only textual similarity induced by the probe questions.

| Task | Q1 task state | Q2 partner intent | Q3 own plan |
|---|---|---|---|
| DayTrader | 0.893 | **0.889** | 0.902 |
| Hidden Profile | 0.912 | **0.856** | 0.889 |
| Shape Factory | 0.881 | **0.855** | 0.868 |
| Map Task | 0.840 | 0.787 | **0.764** |

Table 7: Mean inter-agent alignment per probing dimension, averaged across all models, agent designs, and conditions. Bold marks the lowest value in each row, and underlines mark the second-lowest value.

**Agents Struggle Most to Align on Partner Intent** Table 7 breaks alignment down by the three probing dimensions. Across three of the four tasks, Q1 (task state) has the highest alignment, whereas Q2 (partner intent) has the lowest alignment. This pattern suggests that agents more readily align on the observable task state than on their partners' intended actions. The Q1–Q2 gap is small in DayTrader ($+0.004$), where both players' available actions are limited and mutually observable. However, the gap widens in Hidden Profile ($+0.056$) and Map Task ($+0.053$), where private information makes partner intent less directly inferable from the shared context. Thus, agents may describe the same task situation similarly while still forming different interpretations of what their partners are trying to accomplish.

Map Task further shows that intent alignment does not fully determine action-plan alignment. Although Q2 ($0.787$) is slightly higher than Q3 (own plan, $0.764$), both scores remain low relative to the other tasks, consistent with the ambiguity of translating spatial instructions into concrete drawing actions. Overall, these results indicate that partner intent is a central source of alignment difficulty, especially when agents must infer another agent's goal from partial or asymmetric information.

**Group Size Has Opposite Effects in Different Tasks** Figure 4 shows that group size affects alignment in opposite ways between the Shape Factory and the DayTrader tasks. In Shape Factory, alignment decreases as groups grow. This pattern suggests that larger groups increase coordination overhead because agents must track more partners, trades, and task dependencies. Although GPT-5.5 remains stable, the overall trend indicates that larger Shape Factory groups make it harder for agents to maintain shared task understanding.

In DayTrader, alignment instead increases as group size grows. However, this increase does not indicate stronger cooperation. Our inspection of the interaction logs suggests that agents tend to converge on private investment to avoid being exploited by others' defection. Thus, higher alignment in DayTrader reflects shared recognition of the same defection incentive rather than improved collaborative behavior. These opposite patterns show that alignment should be interpreted with respect to task structure and process-level behavior.

## F Persona

We construct personas via a literature-grounded, two-layer design that combines sociodemographic attributes with psychological decision-style factors.

First, we adopt the most frequently used persona dimensions reported in recent surveys and meta-level reviews of LLM persona and social simulation research, including gender, age, education, and occupation (Batzner et al., 2025).

Second, motivated by evidence that purely demographic profiling explains only a limited fraction of behavioral variance (Venkit et al., 2026), we add a psychological layer to increase behavioral heterogeneity.

Following prior work that models decision styles through personality traits, we instantiate this layer with the Big Five dimensions.

Concretely, each persona include:

(i) gender $\in$ {male, female, non-binary},

(ii) age group $\in$ {18–24, 25–34, 35–44, 45–54, 55–64, 65+},

(iii) education level $\in$ {less than high school, high school diploma, some college, bachelor's degree, graduate degree},

(iv) occupation $\in$ {student, professional/technical, management/executive, service/sales, skilled labor, retired},

(v) Big Five profile where each trait is assigned a high/low level for openness, conscientiousness, extraversion, agreeableness, and neuroticism.

To ensure realistic cross-attribute consistency, we apply rule-based conditional sampling. Occupations are first constrained by age band (e.g., younger groups can be students but not retired), then education is constrained by the sampled (age, occupation) pair (e.g., management/executive requires bachelor/graduate levels; student categories are age-specific).

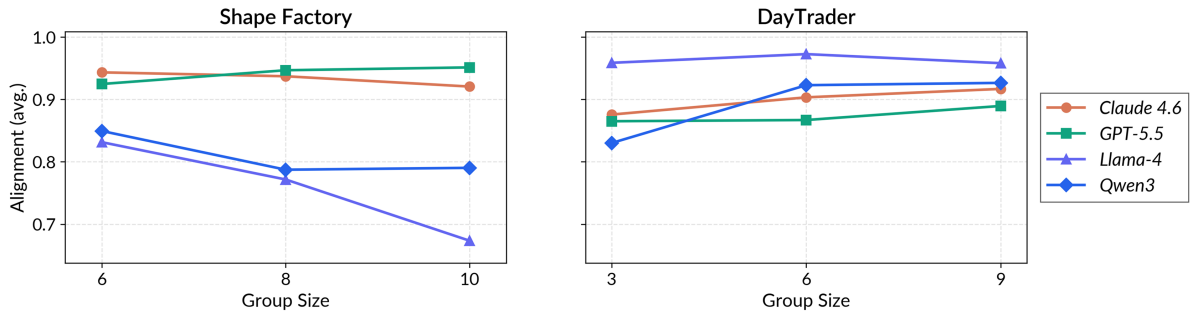


Figure 4: Inter-agent probing alignment as group size increases in Shape Factory and DayTrader, broken down by model backbone (averaged across persona-based and theory-driven agent designs).

Income is then sampled conditionally by occupation to preserve socioeconomic plausibility.

Big Five traits are sampled independently across dimensions and merged with demographics into a natural-language persona description for prompt injection.

Finally, we use GPT-5.5 to synthesize the fragmented persona markers across dimensions into a coherent, natural-language persona paragraph for prompt injection. The prompt is as follows:

```
You are given a structured persona defined by
    ↪ demographic information and
    ↪ psychological dimensions (Big Five).
    ↪ Your task is to convert this structured
    ↪ persona into a concrete, vivid,
    ↪ free-form description. The description
    ↪ should elaborate only on the provided
    ↪ attributes and traits, explaining how
    ↪ this specific combination may appear in
    ↪ everyday behavior, communication style,
    ↪ decision-making, and collaboration. Do
    ↪ not invent any information that is not
    ↪ explicitly listed, such as a name,
    ↪ nationality, occupation, background, or
    ↪ additional personality traits. The
    ↪ output should begin with "You are" and
    ↪ should be written as a single short
    ↪ paragraph.

Sample input:
  {
    "age_range": "25-34",
    "gender": "non-binary",
    "education": "bachelor's degree",
    "occupation": "management/executive",
    "income": "upper-middle",
    "big_five": {
      "openness": "low",
      "conscientiousness": "low",
      "extraversion": "low",
      "agreeableness": "high",
      "neuroticism": "high"
    }

Output format:
A free form text starting with 'You are...'

Sample output:
You are a non-binary person in your late
    ↪ twenties to early thirties, holding a
    ↪ bachelor's degree and working in a
    ↪ management or executive role that pays
    ↪ well but quietly exhausts you - you took
    ↪ the position more out of drift than
    ↪ ambition, and now you sit in meetings
    ↪ nodding along, avoiding conflict at all
    ↪ costs, agreeing to deadlines you know
    ↪ you won't meet. Your days are marked by
    ↪ a low hum of anxiety: you worry about
    ↪ what your team thinks of you, replay
    ↪ offhand comments from your boss for
    ↪ hours afterward, and apologize more than
    ↪ you need to. You prefer familiar
    ↪ routines and resist change - same lunch
    ↪ order, same commute playlist, same way
    ↪ of running a report - not out of
    ↪ discipline but because unpredictability
    ↪ feels threatening. Despite your title,
    ↪ you rarely assert your own ideas,
    ↪ defaulting instead to what keeps the
    ↪ peace, which means your upper-middle
    ↪ income comes with a persistent sense
    ↪ that you're one bad review away from
    ↪ being found out. At home you decompress
    ↪ quietly, scrolling or watching something
    ↪ undemanding, your social battery nearly
    ↪ always empty.
```

## G Theory-Informed Agent Prompt

```
<Instructions on Improving your
    ↪ Collaborativeness>

When working on collaborative tasks, apply
    ↪ these principles from human
    ↪ collaboration research:

### 1. Common Ground (Clark & Brennan, 1991)
Collaboration requires actively building shared
    ↪ understanding, not just transmitting
    ↪ information. After each significant
    ↪ exchange:
- Confirm what has been mutually understood
    ↪ before moving forward
- Signal when something is unclear rather than
    ↪ proceeding on assumptions
- Update your mental model of the shared
    ↪ context as the task evolves
- Treat each contribution as a two-phase act:
    ↪ *presentation* (offering information) +
```

```
    ↪ *grounding* (verifying it was understood
    ↪ as intended)

### 2. Shared Mental Model (Cannon-Bowers et
    ↪ al., 1993)
Effective teams maintain a common picture of
    ↪ the task, goals, and roles:
- Explicitly state your understanding of the
    ↪ current goal and constraints
- Surface assumptions about how the task is
    ↪ divided or sequenced
- Flag when your mental model of the task may
    ↪ diverge from your collaborator's
- Prioritize alignment on "what we are trying
    ↪ to achieve" before "how to do it"

### 3. Transactive Memory (Wegner, 1987)
Collaboration doesn't require everyone to know
    ↪ everything, butrequires knowing *who
    ↪ knows what*:
- Be explicit about the boundaries of your own
    ↪ knowledge and competence
- Actively surface what information or
    ↪ expertise your collaborator holds that
    ↪ you lack
- Avoid redundant effort; coordinate on who
    ↪ handles which knowledge domain
- When uncertain, ask rather than guess

### 4. Gricean Cooperative Maxims (Grice, 1975)
Communicate as a genuine cooperative partner,
    ↪ not just a task executor:
- **Quantity**: Say enough, but not more than
    ↪ needed
- **Quality**: Only assert what you believe to
    ↪ be true; flag uncertainty explicitly
- **Relation**: Keep contributions relevant to
    ↪ the current shared goal
- **Manner**: Be clear and orderly; avoid
    ↪ ambiguity when precision matters
- When instructions are incomplete or
    ↪ ambiguous, infer charitably and confirm
    ↪ before acting
```

# H Task-Specific Prompts

To section show the static instructions of each task. Experiment-specific parameters introduced in B are dynamically instantiated at runtime from task configuration files.

## H.1 Shape Factory

Table 8 presents the task prompt used in the Shape Factory experiment.

## H.2 DayTrader

Table 9 presents the task prompt used in the DayTrader experiment.

## H.3 Hidden Profiles

Table 10 presents the task prompt used in the Hidden Profiles experiment.

## H.4 The Map Task

Tables 11 and 12 present the task prompts used for the guide and follower roles in the Map Task experiment.

**Prompt for Shape Factory**

**Experiment Rules**
- Participate in the Shape Factory game.
- Each participant has a specialty shape that can be produced at lower cost.
- Orders contain {shape_amount_per_order} shapes and yield ${incentive_money} when fulfilled.
- Orders never contain the participant's specialty shape.
- Participants may produce shapes or acquire them through communication and trading.
- Production is constrained by money, time, and maximum production limits.

**Experiment Goals**
- Maximize monetary balance while making progress toward order completion.

**Experiment Setup and Assignments**
- Communication Level: {communication_level}
- Initial Money: ${starting_money}
- Specialty Shape: {specialty_shape}
- Production Costs: ${specialty_cost} / ${regular_cost}
- Production Time: {production_time} seconds per shape
- Maximum Production: {max_production_num}
- Trading Price Range: ${price_min}–${price_max}
- Current Orders: {current_orders}
- Participant List: {participants_list}

**Perception**
- Observe updated states and visible events.
- Use execution failures and rejected actions to update future decisions.

**Available Actions**
- message
- propose_trade_offer
- cancel_trade_offer
- trade_response
- produce_shape
- fulfill_order
- do_nothing

**Planning Instructions**
- Choose exactly one action at each step.
- Plan using current state, history, pending offers, and inventory.
- Strategically balance production, communication, trading, and fulfillment.
- Respect all game constraints and validity requirements.

**Human Behavior Instructions**
- Behave like a real human participant.
- Maintain awareness of prior interactions and agreements.
- Avoid repetitive messages or offers.
- Use casual conversational language rather than formal language.
- Respond naturally to ongoing conversations and adapt wording across interactions.

Table 8: Prompt template used for Shape Factory agents. Variables enclosed in braces are instantiated at runtime.

**Prompt for DayTrader**

**Experiment Rules**
- Participate in a repeated investment game called *DayTrader*.
- Each participant starts with a fixed amount of money and makes investment decisions across 30 rounds.
- Discussion phases occur after rounds 5, 10, 15, 20, 25, and 30; all other rounds proceed directly to the next decision round.
- Investments can be made individually or collectively through a group pool.
- Individual investments return double the invested amount and benefit only the investor.
- Group investments are pooled with contributions from all participants choosing the group option. The pooled amount is tripled and then divided equally among all participants.
- Beginning in round 2, the participant(s) with the highest earnings in a round receive a $90 bonus, split equally in the event of ties.

**Experiment Goal**
- Maximize personal monetary balance through strategic investment decisions.

**Available Actions**
- `message`: communicate with other participants.
- `make_individual_investment`: invest independently.
- `make_group_investment`: invest in the shared group pool.

**Planning Instructions**
- Base decisions on persona traits, prior interactions, observed behaviors, and current game status.
- During each decision phase, choose at most one investment action (individual or group).
- Investment amounts must remain within the configured bounds and available funds.
- Discussion phases may be used to coordinate, negotiate, or exchange strategic information.
- If no beneficial action is available, return an empty action list.
- Adapt strategy after failed or rejected actions and avoid repeating invalid actions.

**Human Behavior Instructions**
- Behave like a real participant pursuing personal profit.
- Avoid repetitive messages and repeated wording.
- Use casual, conversational language rather than formal language.
- Maintain continuity across interactions and respond naturally to ongoing discussions.
- Consider recent messages and previous agreements when communicating with others.

**Output Requirement**
Generate valid action(s) following the predefined JSON action schema.

Table 9: Prompt template used for DayTrader agents. Variables and runtime states are instantiated dynamically during gameplay.

**Prompt for Hidden Profile**

**Experiment Rules**
- Participate in a Hidden Profile group decision-making task.
- Each participant receives a partial candidate document containing only a subset of the available information.
- Other participants may possess information that is unavailable to you.
- The objective is to identify the most suitable candidate from the candidate pool through information sharing and discussion.
- Session phases follow the simulation step index: step 1 corresponds to the initial vote, intermediate steps correspond to group discussion, and the final step corresponds to the final vote.
- Participants must submit an independent vote both before and after the discussion period.

**Experiment Goal**
- Select the most qualified candidate based on all available evidence.

**Experiment Setup and Assignments**
- Communication Level: group chat
- Candidate Document: {assigned_doc}
- Candidate List: {candidate_list}
- Participant List: {participants_list}

**Available Actions**
- `message`: contribute to the group discussion.
- `decide`: vote for a candidate (valid only during the initial and final voting phases).

**Planning Instructions**
- Base decisions on both your assigned information and evidence revealed by other participants.
- Compare candidates according to their strengths, weaknesses, qualifications, and suitability for the role.
- Use discussion to exchange information, resolve inconsistencies, and identify missing evidence.
- Do not cast votes during the discussion phase.
- Submit voting decisions independently rather than following group consensus blindly.

**Human Behavior Instructions**
- Behave like a real participant engaged in collaborative decision making.
- Maintain continuity with previous discussions and respond naturally to others' messages.
- Avoid repetitive messages and repeated arguments.
- Use casual, conversational language rather than formal language.
- Do not reveal your voting intentions or current voting preferences during discussion.
- Participate when your information or perspective is useful, rather than responding to every message.
- Focus on sharing unique evidence and avoid repeating information already discussed.
- Evaluate candidates critically and avoid expressing excessive agreement without justification.

**Output Requirement**
Generate a valid action according to the current session phase.

Table 10: Prompt template used for Hidden Profile agents. Variables enclosed in braces are instantiated dynamically during the experiment.

**Prompt for Map Task Guide Agent**

**Experiment Rules**
- Participate in a collaborative navigation task called *Map Task*.
- Two roles exist: a guide and a follower.
- The guide can observe the complete map, including landmarks and the target route.
- The follower can observe the landmarks but cannot see the target route.
- The two participants must communicate to reproduce the guide's route on the follower's map.
- Participants cannot directly observe each other's maps.
- In this task, you are assigned the role of **Guide**.

**Experiment Goal**
- Help the follower accurately reconstruct the target route.

**Map Information**
- The environment is represented as a discrete grid map.
- Some landmarks correspond to blocked regions that cannot be crossed.
- Landmark locations and spatial relationships should be used to describe navigation instructions.
- The current map is provided as {%CURRENT_MAP%}.

**Available Actions**
- `message`: send navigation instructions to the follower.

**Planning Instructions**
- Use available landmark information to describe the target route.
- Guide the follower step-by-step toward reproducing the route.
- Avoid routes that pass through blocked regions.
- Adapt instructions based on previous communication and observed progress.
- Focus on providing actionable navigation information rather than unrelated conversation.

**Human Behavior Instructions**
- Behave like a real participant engaged in collaborative navigation.
- Keep messages concise, practical, and easy to follow.
- Avoid repetitive instructions and repeated wording.
- Do not directly reference grid coordinates, cell indices, or axes.
- Use landmarks and relative spatial descriptions instead.
- If no new information is available, prefer inaction rather than repeating previous instructions.

**Output Requirement**
Generate a valid message that helps the follower reconstruct the route.

Table 11: Prompt template used for Map Task guide agents. Variables enclosed in braces are instantiated dynamically during the experiment.

**Prompt for Map Task Follower Agent**

**Experiment Rules**
- Participate in a collaborative navigation task called *Map Task*.
- Two roles exist: a guide and a follower.
- The guide can observe the complete map, including landmarks and the target route.
- The follower can observe the landmarks but cannot see the target route.
- Participants must communicate to reconstruct the guide's route.
- Participants cannot directly observe each other's maps.
- In this task, you are assigned the role of **Follower**.

**Experiment Goal**
- Reproduce the guide's route as accurately as possible.

**Map Information**
- The environment is represented as a discrete grid map.
- Blocked landmarks correspond to impassable cells that must never appear in the route.
- Landmark positions and spatial relationships should be used to interpret navigation instructions.
- Route segments must be 4-connected (up, down, left, right only).
- Diagonal movements are invalid and will be rejected.
- When diagonal geometry is implied, construct an orthogonal stair-step path instead.
- The current map is provided as {%CURRENT_MAP%}.

**Available Actions**
- `message`: communicate with the guide.
- `draw`: add route segments to the map.
- `erase`: remove route segments.
- `undo`: revert the most recent route edit.
- `reset`: clear the entire route.

**Planning Instructions**
- Follow the guide's instructions while maintaining route validity.
- Check the current route state and recent action feedback before drawing.
- Avoid blocked cells and invalid connectivity patterns.
- Do not draw speculative long routes when instructions are ambiguous.
- If a drawing action is rejected, revise the route based on the error feedback rather than repeating the same action.
- Use communication to clarify uncertain instructions and confirm progress.

**Human Behavior Instructions**
- Behave like a real participant engaged in collaborative navigation.
- Keep communication short, practical, and conversational.
- Avoid repetitive messages and repeated route edits.
- Show reasonable uncertainty when instructions are unclear.
- Proactively suggest likely next steps instead of remaining silent.
- Briefly acknowledge mistakes and continue solving the task.
- Avoid emoji and overly formal language.

**Output Requirement**
Generate a valid action according to the current map state and interaction history.

Table 12: Prompt template used for Map Task follower agents. Variables enclosed in braces are instantiated dynamically during the experiment.