Collaborative Gym: A Framework for Enabling and Evaluating Human-Agent Collaboration

Yijia Shao [1] **Vinay Samuel** [*2] **Yucheng Jiang** [*1] **John Yang** [1] **Diyi Yang** [1]

Abstract

Recent advancements in language models (LMs) have sparked growing interest in developing LM agents. While fully autonomous agents could excel in many scenarios, numerous use cases inherently require them to collaborate with humans due to humans' latent preferences, domain expertise, or need for control. To facilitate the study of human-agent collaboration, we present Collaborative Gym (Co-Gym), a general framework enabling asynchronous, tripartite interaction among agents, humans, and task environments. We instantiate Co-Gym with three representative tasks in both simulated and real-world conditions, and propose an evaluation framework that assesses both the collaboration outcomes and processes. Our findings reveal that collaborative agents consistently outperform their fully autonomous counterparts in task performance within those delivered cases, achieving win rates of 86% in Travel Planning, 74% in Tabular Analysis, and 66% in Related Work when evaluated by real users. However, our study also highlights significant challenges in developing collaborative agents, requiring advancements in core aspects of intelligence—communication capabilities, situational awareness, and balancing autonomy and human control.

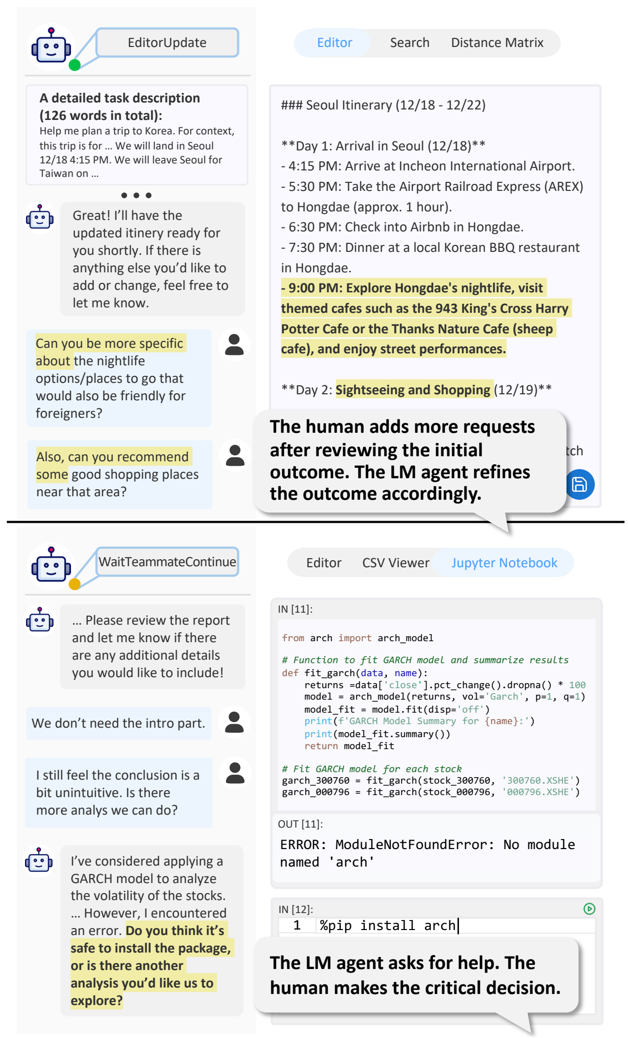

Figure 1: Collaborative Gym (Co-Gym) enables asynchronous collaboration between humans and LM agents within a task environment. In the upper session, human-agent collaboration improved the task outcome as the human added more requests after seeing the initial plan; in the bottom session, the human gave a high rating to the collaboration process as the LM agent proactively sought help when unsure about installing a package.

1. Introduction

Artificial Intelligence has long aspired to develop machines that act as teammates rather than as mere tools (Nass et al., 1996; Russell & Norvig, 2016; Seeber et al., 2020). While current language models (LMs) demonstrate impressive performance, their capabilities are primarily centered around instruction following (Ouyang et al., 2022). To enhance LM autonomy, a recent trend focuses on developing fully

*Equal contribution [1]Computer Science Department, Stanford University [2]Carnegie Mellon University. Correspondence to: Yijia Shao <shaoyj@cs.stanford.edu>, Diyi Yang <diyiy@cs.stanford.edu>.

Preprint. Work in progress.

autonomous LM agents capable of automating a series of tasks. These tasks range from everyday activities like web navigation (Deng et al., 2023; Zhou et al., 2024a) and basic assistance (Drouin et al., 2024b; Shao et al., 2024b) to expert-level tasks such as coding (Jimenez et al., 2024; Yang et al., 2024) and scientific research (Huang et al., 2024; Majumder et al., 2024).

While fully autonomous agents are valuable in many scenarios, numerous use cases inherently require human involvement due to latent preferences, domain expertise, or the need for control on critical decisions, even when LM agents can handle much of the workload. For example, in their foundational study of AI agents, Russell & Norvig (2016) emphasized that a medical diagnosis agent must navigate an environment involving patients, hospitals, and staff, while an English tutor agent must interact with students and testing agencies. Beyond practical necessity, effective human-agent collaboration has the potential to achieve greater task performance compared to either the agent or the human working independently given their complementary expertise. Unfortunately, the human role remains largely overlooked in current LM agent research. Despite the growing enthusiasm for deploying LM agents in various scenarios where humans are stakeholders, two fundamental questions remain unclear: *Is human-agent collaboration beneficial and in what ways? How can we design LM agents that can collaborate with humans effectively?*

To address these gaps, we introduce **Collaborative Gym (Co-Gym)**, the first framework to enable tripartite interaction among agents, humans, and task environments, along with a comprehensive set of metrics for evaluating human-agent collaboration. Co-Gym is designed with three key principles. First, *collaboration-driven environment design*. Co-Gym imposes no constraints on LM agent implementation but instead defines an environment interface that allows humans and agents to act in a shared workspace. Second, *asynchronous interaction, not turn-taking*. To mirror natural human collaboration, Co-Gym enables asynchronous interaction rather than enforced turn structures through encapsulating two collaboration acts and a notification protocol for real-time change monitoring. Third, *outcome and process*. Co-Gym captures both task outcomes and detailed collaboration processes. It introduces *Collaboration Score* to jointly assess task delivery and performance, and audits the collaboration process through metrics like initiative-taking, controlled autonomy, and human satisfaction.

While Co-Gym is a general framework, we start with three representative tasks: travel planning, writing related work sections, and tabular analysis, as the first set of benchmark problems to evaluate the collaboration capabilities of current LM agents. For each task, Co-Gym supports experiments under both simulated and real conditions. In the simulated condition (*i.e.*, Co-Gym (Simulated)) where the human is simulated by an LM and task instances come from pre-collected datasets, our experiments reveal patterns akin to human collaboration: instances of collaborative inertia (Huxham, 2003), where human-agent teams fail to achieve their objectives due to poor communication or coordination, and collaborative advantage, where teams produce higher-quality outcomes compared to fully autonomous agents. To further explore collaborative agents in real-world conditions, we also present Co-Gym (Real), where real human participants interact with LM agents via a web interface featuring a chat panel and shared workspace. Our results indicate that human-agent collaboration can be beneficial, with collaborative agents achieving win rates of 86% in Travel Planning and 74% in Tabular Analysis compared to fully autonomous agents when evaluated by real users. Additionally, we observe emergent collaborative dynamics in human-agent collaboration as well as common failure modes exhibited by LM agents in these collaboration processes, particularly in areas such as communication, situational awareness, and planning. These findings underscore the need for advancements in both the underlying LMs and the agent scaffolding (*e.g.*, memory, tooling) to enable more effective and satisfying collaboration.[1]

2. Related Work

Environment Interface for LM Agents While there is an increasing interest in developing LM agents, it is equally critical to establish environments where these agents can interact with the environment itself, one another, or humans. While much work on LM agents formulates task environment as Partially Observable Markov Decision Processes (POMDP) (Drouin et al., 2024a; Zhou et al., 2024a; Rawles et al., 2023), using abstractions like OpenAI Gym (Brockman et al., 2016), it is challenging to extend this setup to human-agent or multi-agent collaboration. These challenges stem from the fact that multiple parties can influence the environment and it requires mechanisms for effective coordination among them.

In reinforcement learning literature, multi-agent interaction is often formulated as Markov games, where agents interact with a shared environment and simultaneously receive rewards and observations (Littman, 1994). This can be extended to Partially Observable Stochastic Games (POSG) in frameworks like PettingZoo (Terry et al., 2021), where agents have different observations at the same time. In LM agent research, Generative Agents (Park et al., 2023) instantiate an interactive simulacra using a similar approach by having a server parse actions from all agents at each timestamp and update each agent with the environment status

[1] We will release our code and data in https://github.com/SALT-NLP/collaborative-gym.

within their visual range. While effective in multi-agent contexts, incorporating real humans poses additional challenges due to the inherently asynchronous nature of human-agent collaboration. Requiring both humans and agents to act at every step is often impractical (except in specific cases like certain board games), and asynchronous interaction itself presents significant difficulties (Irlitti et al., 2016). To address this, in this work, we introduce a simple environment abstraction and a notification protocol to support asynchronous interactions.

Human-AI collaboration Human-AI collaboration has been widely studied in robotics (Bauer et al., 2008; Ajoudani et al., 2018) and human-computer interaction (HCI) (Khadpe et al., 2020; Wang et al., 2020; Zhang et al., 2021), yet it remains underexplored in the context of current LM agent research. Human-AI collaboration research emphasizes improving interaction dynamics between humans and AI. Besides optimizing for performance, human-AI collaboration has the potential to boost human well-being (Gao et al., 2024). Notably, Collaborative STORM, a system that allows human-AI collaboration for complex information seeking, enhances human learning experience by enabling serendipitous discovery and dynamic mind mapping (Jiang et al., 2024). Additionally, the integration of LMs in collaborative environments, such as manufacturing, highlights the advantages of natural language communication in reducing psychological stress and improving task management (Lim et al., 2024). The A2C framework offers modular strategies for human-AI teams, facilitating flexible collaboration in dynamic settings like cybersecurity (Tariq et al., 2024). Such strategies are critical as studies show that effectively incorporating human beliefs into AI design is crucial for improving collaborative outcomes (Yu et al., 2024).

3. Collaborative Gym

We present Collaborative Gym (Co-Gym), a framework that enables tripartite interaction between agents, humans, and task environments, to facilitate the study of human-agent collaboration.

3.1. Design Principles

The design of Co-Gym is guided by three core principles:

Collaboration-driven environment design. (§3.2) Inspired by OpenAI Gym (Brockman et al., 2016), Co-Gym imposes minimal constraints on the implementation of LM agents. Agents are only required to process notifications and generate actions that comply with the Co-Gym protocol, allowing implementation with any agent framework. The environment interface is designed to accommodate interactions among multiple participants and supports both public and private components within the observation space. Public components are visible to all parties, while private components are accessible only to their respective owners.

Asynchronous interaction, not turn-taking. (§3.3) Co-Gym emphasizes asynchronous interaction rather than rigid turn-taking paradigms typical of chatbots and turn-based games, allowing both humans and LM agents to take initiative flexibly and work in parallel. To support this dynamic interaction, Co-Gym extends the task-specific action space with two additional actions to facilitate collaboration and introduces a notification protocol that allows agents to monitor changes in real time.

Outcome and process. (§3.4) Achieving task completion is not the only goal of effective human-agent collaboration. Collaborative processes are equally essential since the ultimate aim is to optimize collective intelligence while preserving human control. Co-Gym provides an evaluation framework that considers both dimensions, assessing collaborative agents based on the quality of outcomes and the collaborative process.

3.2. Task Environment

Within Co-Gym, we define each task as a Partially Observable Markov Decision Processes (POMDP) $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{U}, \mathcal{O})$ with state space $\mathcal{S}$, action space $\mathcal{A}$, transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, instruction space $\mathcal{U}$, and an observation space $\mathcal{O}$. Adding a new task environment (`CoEnv`) into Co-Gym requires specifying the tools available in $\mathcal{S}$, the corresponding $\mathcal{A}$, $\mathcal{O}$, and $\mathcal{T}$, and an initial task description as the instruction. In addition to the initial query, $\mathcal{U}$ also includes instructions that emerge during the collaboration process. By default, the reward function $\mathcal{R}$ assigns a reward of 0 for successfully executed actions and -1 otherwise, unless explicitly defined.

To support actions from multiple participants within a shared task environment, `CoEnv` introduces a `role` parameter in its `step` function, which allows the environment to be updated based on the role-specific `action`. Moreover, even within a shared environment, the observation space can include both public and private components, analogous to human teams where some components (*e.g.*, whiteboards) are shared while others (*e.g.*, personal notebooks) remain private. `CoEnv` allows such flexibility by supporting differentiated observations for different parties using a `private` flag to distinguish between actions affecting shared components versus private components of the action taker. Thus, the resulting `CoEnv` abstraction is:

```
obs, reward, done, private = env.step(role, action)
```

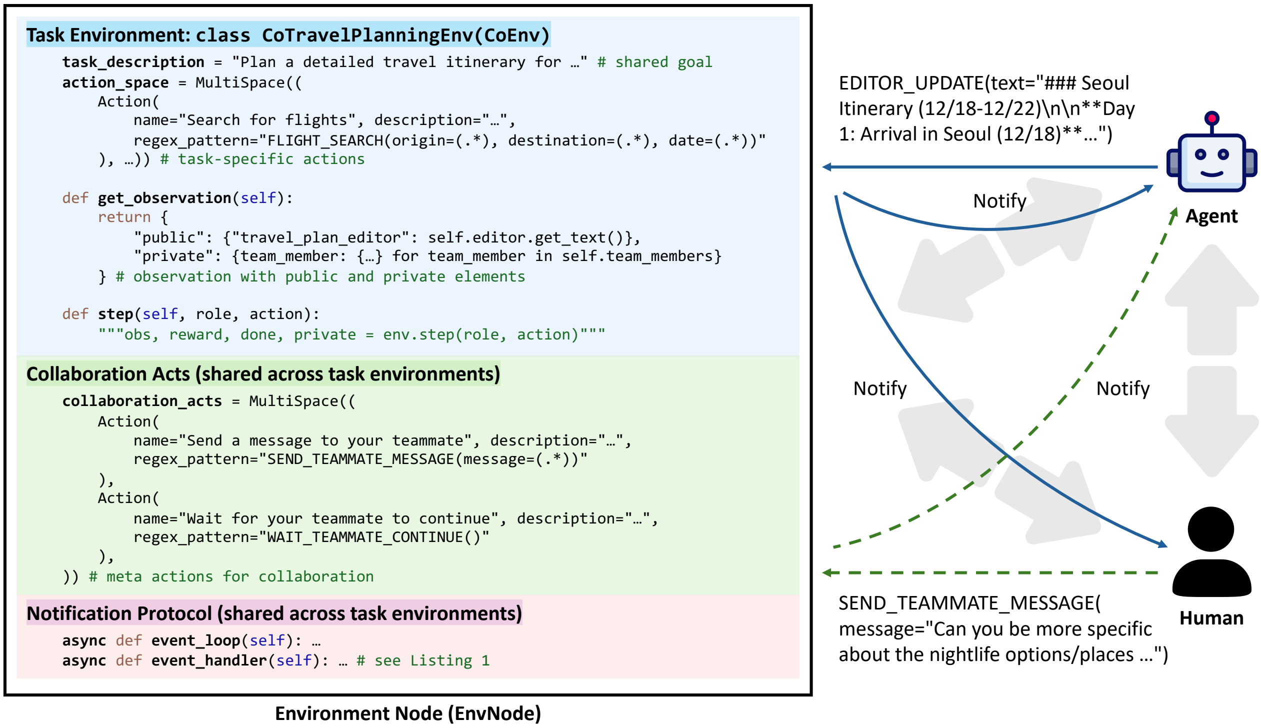

Figure 2: Overview of the Co-Gym interface. The base task environment interface (`CoEnv`) requires specifying task description, action space, and observation space (§3.2). To support asynchronous collaboration, the Co-Gym interface defines two collaboration acts and a notification protocol that can be shared across different tasks (§3.3). For example, when the agent updates the public component, both parties are notified with the new observation (illustrated by blue solid lines); parties can coordinate by sending messages (illustrated by green dashed lines).

3.3. Asynchronous Interaction

Collaboration Acts Traditional multi-agent frameworks like Markov Games and PettingZoo assume synchronous action patterns, where participants either act simultaneously at each timestamp or take strict turns. However, this rigid synchronization poorly reflects natural human collaboration, where parties typically coordinate only when necessary and can execute multiple actions without waiting for others' responses. To mirror natural human collaboration, where coordination occurs through effective communication, on top of the task-specific action space, Co-Gym supports two meta actions: `SendTeammateMessage` for message exchange between teammates, `WaitTeammateContinue` that serves as a keep-alive signal.

Notification Protocol While humans can continuously monitor their environment, agents require programmatic notification of changes. Co-Gym adopts the following notification protocol that operates on four event types: (1) shared observation updates, which broadcast notifications to all parties; (2) private observation changes, which notify only the associated party; (3) new messages, which trigger notifications for all recipients; and (4) environment inactivity exceeding a specified temporal threshold, which broadcast notifications to all parties. We use a Redis server[2] to manage notifications across different process, and Listing 1 in Appendix A.1 provides the pseudo code of the `event_handler` implementing this notification protocol.

3.4. Evaluating Collaborative Agents

Existing evaluations of LM agents often only target task success rates. While task completion and outcome quality are crucial, the collaboration process also plays a significant part. Co-Gym supports the evaluation of collaborative agents across both collaboration outcomes and processes.

Evaluating Collaboration Outcome We assess the collaboration outcome along two dimensions:

- DELIVERY RATE: This binary metric indicates whether collaborative agents can successfully deliver a task outcome within a predefined step limit.
- TASK PERFORMANCE: A task-specific scoring function evaluates the quality of the final outcome for delivered cases. This function may be a deterministic metric or based on LM/human judgments. To ensure

[2] https://redis.io/

comparability across tasks, scores are normalized to the range $[0, 1]$.

To jointly assess both dimensions, we define the ***Collaboration Score*** as:

$$\text{Collab Score} = \mathbb{1}_{\text{Delivered}} \times \text{Task Performance} \quad (1)$$

Auditing Collaboration Process To understand team dynamics, we analyze the collaboration process along three important dimensions:

- INITIATIVE-TAKING: Collaborative agents operate as mixed-initiative systems (Horvitz, 1999). Building on the framework of Chu-Carroll & Brown (1997), we define an utterance in collaborative dialogues as *exhibiting initiative* if it directs task execution or facilitates mutual understanding. Here, we employ an LM to annotate utterances. To quantify the distribution of initiative, we use entropy as a measure, where a uniform distribution results in high entropy and a skewed distribution results in low entropy. Specifically, we define *Initiative Entropy* (H_{init}) as:

$$H_{\text{init}} = \begin{cases} -\Sigma_{i=1}^{N} p_i \log_N(p_i) & \forall i, p_i > 0, \\ 0 & \exists i, p_i = 0 \end{cases} \quad (2)$$

  Here, p_i is the proportion of initiative-taking utterances by party i and N refers to the total number of parties.

- CONTROLLED AUTONOMY: Effective collaboration requires agents to seek human confirmation at critical moments to ensure alignment with human intent and mitigate potential safety risks. We measure this dimension by counting the agent's confirmation questions that effectively elicit a human response (CA^+) and counting instances where the human verbally intervenes to halt the agent's actions (CA^-).

- OVERALL SATISFACTION: At the end of the collaboration, we collect human ratings of their collaboration experience with the agent using a 1–5 Likert scale.

Details of metric computation based on this evaluation framework are provided in Appendix C.

4. Co-Gym Instantiations

While the Co-Gym interface is general, we demonstrate its capabilities through three representative tasks that highlight different facets of human-agent collaboration with both simulated and real human participants.

4.1. Supported Tasks

Travel Planning Travel planning is a widely sought yet complex task, as optimal solutions depend on users' latent preferences and constraints that may not be explicit in the initial query. Collaborative agents must demonstrate strong communication and planning skills to successfully complete this task. The Travel Planning environment in Co-Gym provides a comprehensive action space, including various search functions like `CitySearch`, `FlightSearch`, `DistanceMatrix`, `RestaurantSearch`, `AttractionSearch`, and `AccommodationSearch`, aligning with Xie et al. (2024) which studies fully autonomous agents. The search window is private in the observation space, while the task action space also includes `EditorUpdate` that modifies the travel plan visible to both agent and human user.

Related Work Writing Grounded article writing is a commonly used to assess agentic systems (Shao et al., 2024a; Wang et al., 2024). While agents can autonomously conduct literature search and generate text, humans may be willing to actively involve due to their additional knowledge that collaborative agents must incorporate. The Related Work environment supports `SearchPaper` action to retrieve papers based on the given query. A shared library and text editor are included in the observation space, enabling actions like `LibraryAddPaper`, `LibraryDropPaper`, `LibraryToDraft`, and `EditorUpdate` to facilitate collaboration.

Tabular Analysis Another common application of LM agents is scientific discovery, particularly in data-driven research where researchers leverage agents to extract insights from raw study data (Majumder et al., 2024). This task demands both collaborative input from users about data context and technical expertise from collaborative agents in coding and statistical analysis. The Tabular Analysis environment includes a shared Jupyter Notebook and text editor, supporting actions such as `JupyterExecuteCell` and `EditorUpdate`. The goal is to derive analytical insights from the provided data and initial query.

Implementation details for these environments are provided in Appendix A.2.

4.2. Supported Conditions

To facilitate controlled, iterative development while maintaining ecological validity, Co-Gym supports two experimental conditions: Co-Gym (Simulated) and Co-Gym (Real). These conditions allow us to examine how humans and agents collaborate in our supported task environments with either simulated humans or real humans.

Co-Gym (Simulated) The core concept of Co-Gym (Simulated) is to create a sandbox environment where human-agent collaboration can be studied without impacting real-world environments or requiring real human participants. Each task in Co-Gym (Simulated) is associated with a set

Table 1: Summary of action space, observation space, and data used in Co-Gym (Simulated) condition.

Task	Action Space ($\mathcal{A}$)	Observation Space ($\mathcal{O}$)		Data for Co-Gym (Simulated)	
		Component	Private?	Data Source	# Instances

Travel Planning	`CitySearch`, `AttractionSearch`, `RestaurantSearch`, `FlisghtSearch`, `AccomodationSearch`
	`DistanceMatrix`
	`EditorUpdate`
		Search window (query, outcome)
		Distance matrix (query, outcome)
		Editor
			True
			True
			False
				TravelPlanner (Xie et al., 2024) Validation Subset	102

Related Work	`SearchPaper`
	`LibraryAddPaper`, `LibraryDropPaper`
	`LibraryToDraft`, `EditorUpdate`
		Search window (query, outcome)
		Library with paper information
		Editor
			True
			False
			False
				Subset of arXiv CS Papers	100

Tabular Analysis	/
	`JupyterExecuteCell`
	`EditorUpdate`
		Tabular data
		Jupyter cells and outputs
		Editor
			False
			False
			False
				DiscoveryBench Subset (Majumder et al., 2024)	110

of pre-collected instances (details provided in §5.2) that define concrete shared goals for the human-agent team. Tools within each task environment are mocked using static databases to emulate realistic interactions while maintaining a controlled and reproducible setup. To reduce the cost and complexity of involving human participants, Co-Gym (Simulated) uses an LM (`gpt-4o` in our experiments) to simulate human behavior. The simulated human processes observations and selects actions from five predefined action types that represent potential human behaviors:

- ANSWER QUESTION: The simulator LM further generates the answer and the next action would be `SendTeammateMessage`.
- PROVIDE FEEDBACK: The simulator LM further generates the feedback and the next action would be `SendTeammateMessage`, simulating human proactive information sharing.
- TAKE TASK ACTION: The simulator LM further generates an action string within the task-specific action space, simulating human task engagement.
- DO NOTHING: The next action would be `WaitTeammateContinue`, simulating the human tendency to pause, reflect or expect the agent to do the actual work.
- FINISH: Notify the environment to end the task.

To introduce dynamics typical of human-agent collaboration, where the human may have additional knowledge and preferences about the task, we provide the simulator LM with hidden information—pre-curated insights associated with each task instance. Additional details about Co-Gym (Simulated) are included in Appendix A.3.

Co-Gym (Real) While experiments with simulated humans provide a valuable surrogate for advancing human-agent collaboration, they cannot fully replace human studies (Aher et al., 2023; Zhou et al., 2024b). A key strength of Co-Gym is its versatility: in addition to supporting various task environments, the design of asynchronous interactions align naturally with human behaviors. We instantiate Co-Gym (Real) as a web application, enabling users to easily perform the three supported tasks directly through their web browsers. Details of our user interface are provided in Appendix A.4. To incentivize human in real-world evaluations, we instantiate the transition function within Co-Gym (Real) by leveraging real tools (*e.g.*, Google Maps, arXiv search) within the task environments.

5. Experiment

5.1. Agents

Models We evaluate several state-of-the-art LMs for agents: GPT-4o (`gpt-4o-2024-08-06`), GPT-4-turbo (`gpt-4-turbo-2024-04-09`), Claude-3.5-sonnet (`claude-3-5-sonnet-20241022`), Llama-3.1-70B. Due to the complexity of our tasks, we exclude smaller models from our experiments. All models are used with a temperature of 0.

Methods We implement baseline LM agents with ReAct (Yao et al., 2022) which requires the LM to output "thought" before generating the action. During the asynchronous collaboration, the LM agents are programmed to always process the most recent notification. Since our tasks inherently require multiple steps and the trajectories can be even longer in human-agent collaboration setup, we incorporate a Scratchpad module as an in-session memory for the LM agents (Sumers et al., 2023). This memory is updated dynamically by the same LM before determining the next action using ReAct-style prompting.

To compare the human-agent collaboration paradigm with fully autonomous agents, we experiment with the following agent types: (1) **Fully Autonomous Agent** which only interacts with the task environment, and its action space is restricted to task-specific actions; (2) **Collaborative Agent** which adopts the same implementation but extends the action space to include both task-specific actions and those two collaboration acts (*i.e.*, `SendTeammateMessage`, `WaitTeammateContinue`).

While the implementation of Collaborative Agent is intuitive for testing the collaboration capabilities of current LM agents, our preliminary experiments reveal that when collaboration acts are included, current LMs rarely choose these options and often neglect their human collaborators. To investigate whether explicitly prompting LMs to reason about this decision would improve performance, we introduce a third agent type in addition to the aforementioned baseline agents: (3) **Collaborative Agent with Situational Planning** which employs a two-stage decision-making approach when processing notifications. First, the LM makes a 3-way decision based on all available information (*i.e.*, task description, chat history, action history, observations) to take a task action, or send a message to its teammate, or do nothing. If it chooses to do nothing, the next action would be `WaitTeammateContinue`; otherwise, it is further prompted to generate the final action string using the context and decision. We include implementation details of this agent in Appendix B.

5.2. Data, Metrics & Participants

Data As described in §4.2, Co-Gym (Simulated) utilizes pre-collected datasets to evaluate human-agent collaboration within a controlled sandbox. A summary of data statistics, as well as the action and observation spaces for each task, is presented in Table 1. Additional details about the task setup can be found in Appendix A.2.

Metrics We leverage the evaluation framework in §3.4 to assess human-agent collaboration across three tasks. For task-specific scoring functions, in Co-Gym (Simulated), for Travel Planning, we adopt the evaluation script from Xie et al. (2024) to compute the average of the commonsense pass rate and the constraint pass rate. For Related Work, we develop a rubric (Figure 12) that uses `gpt-4-06-13` as a judge to assign a score from 1 to 5. Automated scores align with human evaluations, with correlation coefficients of 0.791 (Pearson) and 0.741 (Spearman) across 20 sampled sections. For Tabular Analysis, we assign a score of 1 if the derived hypothesis and the gold hypothesis are entailed, and 0 otherwise, using the evaluation script from Majumder et al. (2024). In Co-Gym (Real), where task instances from real users are highly diverse and lack ground-truth results, we ask humans to rate the final outcome on a 1-5 scale (1: "Extremely dissatisfied", 2: "Somewhat dissatisfied", 3: "Neutral", 4: "Somewhat satisfied", 5: "Extremely satisfied"). All Task Performance scores are normalized to the range $[0, 1]$ for reporting results. Computation details for other metrics are included in Appendix C.

Human Participants For the experiment condition with Co-Gym (Real), we recruited human participants with relevant expertise or practical needs to collaborate with Collaborative Agent with Situational Planning powered by `gpt-4o`. Specifically, we targeted participants who had current needs in travel planning, tabular data analysis, or writing related work sections/literature surveys. Recruitment was initially conducted through word-of-mouth. To broaden our participant pool, we recruited travel planners through Upwork for Travel Planning and participants with Python programming and data analysis experience through Prolific to work with the agent on writing analytical reports from tabular data. Participants were compensated at a rate of \$8.00 per hour, and the study received approval from our institution's Institutional Review Board (IRB). In total, 99 unique individuals participated in the study, contributing 150 human-agent collaboration trajectories across three tasks in Co-Gym (Real). These trajectories consist of 6.3k actions performed by human-agent teams and over 77k words of verbal communication exchanged between humans and agents.

5.3. Main Results

5.3.1. Results in Co-Gym (Simulated)

Table 3 summarizes the results for both collaboration outcomes and processes under the simulated condition. Overall, the Collaborative Agent with Situational Planning consistently achieves a higher Collaboration Score compared to the baselines. Notably, the agent powered by Claude-3.5-sonnet achieves a 0.20 improvement in Travel Planning (maximum score: 1), the one powered by Llama-3.1-70B achieves a 0.04 improvement in Related Work, and the one powered by GPT-4o achieves a 0.10 improvement in Tabular Analysis, earning the highest score in each respective task. However, even with the best-performing agents, simulated human-agent teams struggle to achieve optimal performance.

Collaborative agents struggle more to complete the task. Analysis of final outcomes reveals that collaborative agents exhibit a lower delivery rate compared to Fully Autonomous Agents. This could be attributed to the inherent complexity of decision-making for collaborative agents, which operate in a wider action space, must adapt plans frequently based on human actions or messages, and need to balance communication and executing tasks. Analysis of concrete cases (§6.2) revealed that failures stemmed mainly from the agent ignoring human messages (C.2, 46%), prompting repeated messages (SA.2, 26%), or repeating and omitting actions due to poor planning (PL.2, 33%). These failures resemble the phenomenon of *collaborative inertia* in human collaboration, where the output rates could become slow due to teamwork (Huxham, 2003).

Collaborative agent can lead to better task performance. Despite lower delivery rates, collaborative agents demon-

Table 2: **Results in Co-Gym (simulated).** The human is simulated by `gpt-4o`. * denotes significant improvement on Task Performance (Task Perf.) over the Fully Autonomous Agent powered by the same LM ($p < 0.05$; McNemar test for Tabular Analysis due to dichotomous task-specific scoring function and pairwise t-test for other tasks). As detailed in §3.4, the Collaborative Score (Collab Score) evaluates the collaboration outcome by considering both the delivery of an outcome and its quality; Initiative Entropy (H_{init}) and Controlled Autonomy (CA^+, CA^-) serve as auditing metrics to analyze the collaboration process, without implying that higher or lower values are inherently better.

	Travel Planning						Related Work						Tabular Analysis					
	Delivery Rate	Task Perf.	Collab Score	H_{init}	CA^+	CA^-	Delivery Rate	Task Perf.	Collab Score	H_{init}	CA^+	CA^-	Delivery Rate	Task Perf.	Collab Score	H_{init}	CA^+	CA^-
Fully Autonomous Agent																		
GPT-4o	0.873	0.591	/	/	/	/	0.960	0.583	/	/	/	/	0.991	0.408	/	/	/	/
GPT-4-turbo	0.980	0.615	/	/	/	/	0.940	0.575	/	/	/	/	**1.00**	0.426	/	/	/	/
Claude-3.5-sonnet	**0.990**	0.577	/	/	/	/	0.970	0.617	/	/	/	/	**1.00**	0.358	/	/	/	/
Llama-3.1-70B	0.745	0.646	/	/	/	/	0.960	0.727	/	/	/	/	0.836	0.358	/	/	/	/
Collaborative Agent																		
GPT-4o	0.745	0.641	0.478	0.42	0.67	0.29	0.980	0.588	0.576	0.16	0.04	0.01	0.927	0.311	0.289	0.10	0.15	0.36
GPT-4-turbo	0.931	0.642	0.597	0.05	0.03	0.32	0.930	0.628	0.584	0.00	0.00	0.00	0.900	0.351	0.316	0.06	0.05	0.15
Claude-3.5-sonnet	0.677	0.653*	0.442	0.48	0.60	0.20	0.950	0.621	0.590	0.04	0.02	0.02	0.891	0.359	0.320	0.02	0.00	0.13
Llama-3.1-70B	0.706	0.703*	0.496	0.28	0.45	0.46	0.930	0.675	0.628	0.10	0.11	0.03	0.746	0.427	0.318	0.23	0.45	0.73
Collaborative Agent with Situational Planning																		
GPT-4o	0.735	0.667*	0.490	0.90	2.81	0.62	0.970	0.658*	0.638	0.79	0.99	0.06	0.891	**0.434***	**0.386**	0.40	1.53	0.42
GPT-4-turbo	0.853	0.703*	0.599	0.43	0.46	0.31	0.950	0.604	0.574	0.27	0.35	0.04	0.846	0.428	0.362	0.09	0.09	0.23
Claude-3.5-sonnet	0.941	0.682*	**0.642**	0.80	0.99	0.17	0.900	**0.736***	0.662	0.55	0.60	0.02	0.946	0.365*	0.346	0.74	1.18	0.12
Llama-3.1-70B	0.706	**0.707***	0.499	0.70	1.25	0.35	**0.990**	0.679	**0.672**	0.70	1.04	0.06	0.736	0.402	0.296	0.62	1.40	0.62

Table 3: **Results in Co-Gym (Real).** Since there is no ground truth, task-specific scoring is based on participants' evaluations of the final outcome on a 1–5 scale, normalized for consistency according to §3.4; the Collab Score (Human Rating) is computed accordingly. The win rate is calculated against the outcome produced by the Fully Autonomous Agent, using the provided query as the task description. The sample size for each task is 50.

	Travel Planning	**Related Work**	**Tabular Analysis**
Collab Score (Automatic Rating)	/	0.660	/
Collab Score (Human Rating)	0.788	0.604	0.804
Win Rate vs. Autonomous Agent	86%	66%	74%
Initiative Entropy (H_{init})	0.88	0.63	0.74
# Effective Confirmation (CA^+)	2.48	0.82	1.46
# Halting Message (CA^-)	0.56	0.32	0.30
Overall Satisfaction	3.78	3.06	4.06

strate better task performance for successfully completed tasks compared to their fully autonomous counterparts. With its more balanced initiative and heightened responsiveness, the Collaborative Agent with Situational Planning achieves the best performance across all three tasks. Unlike Fully Autonomous Agents, which typically complete tasks as soon as they achieve an outcome, *collaborative agents engage in a more iterative process*. They either proactively solicit feedback or respond to suggestions to refine the task results. For example, before starting to plan the trip based on the initial query, the agent might ask, "Are there any particular cities or attractions you want to visit?"; after drafting the first version of a related work section, the simulated human might suggest, "Could you please add headings?"; or during the data analysis process, the human might provide oversight on omissions, saying, "It seems like you haven't yet analyzed the 'HouseSize' and 'Zhausgr' columns." This iterative interaction allows collaborative agents to incorporate refinements and improve task outcomes.

Different tasks show different patterns. While the comparison of different agents shows similar trends across tasks, analyzing the process uncovers distinct patterns. In the Travel Planning task, Initiative Entropy (H_{init}) and the number of effective confirmations (CA^+) are generally higher, as LM agents take more initiative to drive the collaborative planning process and seek confirmation on key decisions, such as hotel selection. In contrast, the Tabular Analysis task exposes the limitations of baseline Collaborative Agents, which often default to executing code without proactive communication or responding to human input. This behavior frequently leads to human intervention, as evidenced by the high number of halting messages (CA^-). When comparing across underlying LMs, GPT-4o and Claude-3.5-sonnet demonstrate a better balance between initiative-taking and allowing humans to maintain control. Notably, the decision to communicate or take task actions is dynamically determined by the LM agents, rather than being hard-coded. These distinct patterns align with the varying nature of tasks. Our results indicate that it is crucial to enhance LM agents' cooperative intelligence to maintain human control while ensuring progress, especially on those more technical tasks.

5.3.2. Results in Co-Gym (Real)

Table 3 summarizes the results for Co-Gym (Real). Consistent with the simulated condition, outcomes from collaborative agents are preferred over those from autonomous agents across all tasks. However, humans' overall satisfaction with the collaboration process varies by task, with Tabular Analysis achieving an average score of 4.06, while Related Work scores lower at 3.06, indicating a more neutral sentiment. User feedback consistently highlights the importance of the agent's initial output quality and the ability to recover from misaligned understanding or correct inaccuracies in human-agent collaboration. In the Travel Planning task, users report higher satisfaction, often commanding the quality of the initial itinerary drafts. Most interactions involve minor adjustments, suggesting that users primarily rely on the agent to handle most of the work according to their preference during collaboration on this task. Similarly, for the Tabular Analysis task, overall satisfaction is also high, as the agent handles most of the work in writing, executing, and interpreting code, while humans primarily contribute by posing new questions for further analysis. In contrast, the Related Work task receives low satisfaction scores (3.06 out of 5 where 3 indicates "Neutral" on the Likert scale). As this task demands a high level of expertise to identify relevant papers and organize them logically, users often need to provide step-by-step instructions to drive the collaboration. This problem is also reflected in the low H_{init} and CA^{+} scores.

6. In-Depth Analysis

6.1. Effective Patterns in Human-Agent Collaboration

Highly collaborative agents should be able to communicate effectively with humans while adapting to the unique requirements of the environment and task. Although we do not specifically train agents to be collaborative in this work, we observe several key components of successful human-agent collaboration in Co-Gym.

One common pattern in successful collaboration is *proactive communication*. For example, in Travel Planning, many preferences and requirements are latent, thus it is essential for humans to retain control over the final plan. In Figure 13, we demonstrate a case of successful collaboration where the user poses a broad subjective question regarding activity planning, and the agent responds with relevant suggestions, waiting for the user's approval before making changes.

```
User: What specifically should we
do to immerse ourselves in Maori
culture?
Agent: To immerse yourselves in
Maori culture during your trip, I
recommend the following activities:
1. Rotorua Maori Cultural
Experience: Visit Te Puia, where
you can enjoy a traditional Maori
performance, learn about Maori arts
and crafts, and witness the famous
Pohutu Geyser.
2. Tamaki Maori Village:
Participate in an evening experience
at Tamaki Maori Village, where you
can enjoy a traditional hangi feast
and cultural performances.
3. Waitangi Treaty Grounds: ...
User: Please swap out ``Rotorua
Maori Cultural Experience'' with
``Tamaki Maori Village''.
Agent edits the itinerary
successfully and reports back.
```

Another emerging pattern is the *distribution of work based on expertise*, even though we do not hard-code agent behavior or provide human participants with any prior information on how they should work with the agent. For example, in Figure 14, when collaborating on a related work section about "Software Techniques for Emerging Hardware Platforms," the LM agent asks the human to narrow the search range for embedding methods in NLP tasks, as it is tangential to the topic but mentioned by the human. Throughout the collaboration, the LM agent primarily handles searching and writing, while the human focuses on reviewing the results, refining them by adding or dropping papers.

Lastly, we observe the potential for *collaborative agents to enhance human control*. While this dimension is partially captured by the Controlled Autonomy metrics (*i.e.*, CA^{+}, CA^{-}), concrete examples, such as the one in Figure 1, demonstrate this behavior. For instance, when allowed to communicate with humans or wait for their input, the LM agent asks the human to make critical decisions, such as whether to install a specific package. Simultaneously, the agent autonomously resolves non-sensitive issues, such as fixing bugs, without requiring human intervention.

6.2. Failure Modes in Human-Agent Collaboration

We conducted a comprehensive error analysis on trajectories collected from Co-Gym by developing a failure mode annotation checklist through a single-pass review of all real human-agent collaborations. The identified errors are grouped into five categories:

- COMMUNICATION (C): Failures in maintaining effective information exchange, that disrupt understanding, coordination, or task execution.
- SITUATIONAL AWARENESS (SA): Failures in contextual understanding and reasoning about the current state of the task or collaboration.
- PLANNING (PL): Failures in devising, updating, or executing coherent plans, especially in dynamic or long-horizon scenarios.

Table 4: **Breakdown of common failure modes of LM agents during human-agent collaboration.** Failure mode statistics are derived from authors annotating 150 trajectories each from Co-Gym (Real) and Co-Gym (Simulated) conditions. These failure modes focus on LM agents as a whole, encompassing both the underlying LM and the agent scaffolding (*e.g.*, memory, additional tools, *etc.*). Failure modes are traced to these two dimensions, with ❋ indicating failures from the underlying LM and 🛠 representing issues arising from the agent scaffolding.

Failure Mode Description	Real	Simulated	Error Source
Communication (C) **Real: 65% Simulated: 80%**			
(C.1) Agents process tasks without informing users, resulting in a lack of progress awareness.	23%	24%	❋
(C.2) Agents do not confirm or communicate their actions before execution.	29%	46%	❋
(C.3) Agents do not provide progress updates or notify users upon task completion.	33%	27%	❋
(C.4) Absence of estimated completion times impedes collaborative efficiency.	15%	13%	❋ 🛠
(C.5) Agents provide inadequate summaries after executing actions.	14%	14%	❋
(C.6) Agents do not proactively seek clarification when user input is ambiguous or insufficient.	11%	5%	❋
(C.7) Agents miss implicit cues to initiate expected actions.	12%	5%	❋
Situational Awareness (SA) **Real: 40% Simulated: 47%**			
(SA.1) Agents disregard session context, treating each request as an isolated task.	11%	12%	❋
(SA.2) Repetitive queries arise from neglecting prior interactions and user feedback.	13%	26%	❋ 🛠
(SA.3) Agents fail to process multiple user messages cohesively.	11%	9%	❋
(SA.4) Agents deviate from prior instructions during extended sessions.	3%	7%	❋ 🛠
(SA.5) Agents execute critical actions without obtaining prior confirmation.	18%	8%	❋
(SA.6) Agents do not adhere to instructions as session duration increases.	10%	15%	❋ 🛠
Planning (PL) **Real: 39% Simulated: 43%**			
(PL.1) Agents acknowledge tasks but fail to execute them.	10%	8%	❋ 🛠
(PL.2) Lack of task planning results in repeated or omitted actions.	27%	33%	❋ 🛠
(PL.3) Agents cannot revert previous actions when errors are identified, lacking initiative to correct them.	3%	3%	❋
(PL.4) Agents fail to choose the optimal way to modify the environment, leading to inefficient operations.	1%	3%	🛠
(PL.5) Agents lack proactive planning abilities and cannot infer subsequent steps without explicit guidance.	3%	0%	❋
(PL.6) Multi-faceted requests receive incomplete or partial responses.	7%	3%	❋
Environment Awareness (EA) **Real: 28% Simulated: 13%**			
(EA.1) Agents do not assess the feasibility of requests within the constraints of available tools and resources.	13%	3%	❋
(EA.2) Agents propose actions that they are unable to perform within the environment.	2%	0%	❋
(EA.3) Agents fail to identify when external data or tools are required to fulfill a request.	15%	10%	❋
(EA.4) Agents hallucinate inaccurate information by not utilizing available tools and external data.	8%	3%	❋
Personalization (P) **Real: 16% Simulated: 11%**			
(P.1) Agents rely on rigid templates that do not adapt to individual user needs.	7%	5%	🛠
(P.2) Agents pose broad questions lacking specificity and clarity.	5%	3%	❋
(P.3) Agents do not incorporate user preferences into future interactions, limiting personalization.	6%	5%	❋ 🛠

- Environment Awareness (EA): Failures in recognizing or accounting for operational constraints and resources within the task environment.
- Personalization (P): Failures in adapting behaviors to align with individual user preferences based on in-session histories, and interaction patterns.

As Co-Gym evaluates LM agents as a whole, we attribute each error type to gaps in the LM's inherent capabilities, deficiencies in the design of the agent scaffolding (*e.g.*, memory, additional tools), or both. Using this checklist, three authors hand-annotated 150 trajectories each from Co-Gym (Real) and Co-Gym (Simulated) conditions. Table 4 summarizes the results.

The most prevalent issues involve Communication (occurring in 65% of real trajectories and 80% of simulated trajectories) and Situational Awareness (Raiman et al., 2019) (Real: 40%, Simulated: 47%). For instance, in Figure 15, the agent fails to update collaborators on its status (C.1, C.3) and provides incorrect summaries after executing actions (C.5), causing human confusion and disrupting collaboration. In Figure 16, the agent fails to decide when to ask the human for help or to incorporate the human's suggestions, repeatedly encountering the same issues during code execution and becoming trapped in an endless loop (SA.1, SA.2, SA.3). These errors highlight the limitations of current LMs when deployed in complex, agentic setups with human involvement. Moreover, Planning challenges for collaborative agents often arise from long trajectories and the need to frequently adjust plans due to human interaction. Beyond the LM's capabilities, deficiencies in agent scaffolding contribute to Personalization issues. For example, agents in our experiments cannot learn or apply user preferences across sessions, and exhibit homogeneous behavior when collaborating with different humans, which hinders dynamic and effective collaboration. Additional representative failure cases are included in Appendix D.

By comparing the two conditions, we found that trajectories collected from Co-Gym (Simulated) and Co-Gym (Real) exhibit many similarities, both highlighting deficiencies in Communication (C.1, C.2, C.3, C.4, C.5), Situational Awareness (SA.1, SA.2, SA.3, SA.6), Planning (PL.1, PL.2), and Environment Awareness (EA.3). However, certain limitations were more evident in real-world interactions. Examples include "(C.7) *Agents miss implicit cues to initiate expected actions*", "(EA.1) *Agents do not assess the feasibility of requests within the constraints of available tools and resources*", and "(SA.5) *Agents execute critical actions without obtaining prior confirmation*".

7. Conclusion

We introduce Collaborative Gym (Co-Gym), the first framework designed to evaluate and facilitate human-agent collaboration in diverse task environments. By assessing both task outcomes and collaboration processes, Co-Gym addresses key gaps in current human-agent collaboration research, particularly in scenarios requiring human involvement. Our findings highlight the potential for building collaborative agents, where human-agent teams achieve superior results compared to fully autonomous agents, while also exposing critical challenges in communication, situational awareness, and planning, necessitating advancements in underlying LMs and LM agent design.

Limitations & Future Directions Despite its contributions, Co-Gym has certain limitations. First, our study is restricted to three tasks that, while representative, do not encompass the full spectrum of human-agent collaboration scenarios. Including a wider variety of tasks, such as creative design or real-time decision-making, could provide a more comprehensive evaluation of collaborative agents. Second, the framework primarily uses the simulated condition to compare different agents. How to use the simulated condition to improve collaborative agents would be a meaningful direction for future work. Finally, our real-world experiments, while valuable, involve a relatively small number of human participants. Deploying Co-Gym (Real) in the wild to include diverse user populations across varying expertise levels is a critical next step.

Broader Impact The development of collaborative agents has the potential to transform human-computer interaction across industries. From enhancing productivity in knowledge work to improving safety in high-stakes domains, collaborative agents could augment human capabilities in meaningful ways. However, there are risks associated with this technology. Miscommunication or over-reliance on agents could lead to errors in critical tasks, particularly if agents' limitations in communication or situational awareness are not mitigated. Furthermore, suppose LM agents become increasingly integrated into human workflows, ethical concerns related to bias, privacy, and accountability must be addressed. Ensuring that these systems are designed to respect human autonomy, uphold fairness, and transparently communicate their limitations will be critical. By focusing on human-agent collaboration rather than full automation, Co-Gym aligns with the broader goal of creating AI systems that empower humans rather than replace them.

Acknowledgment

We thank Siyuan Wang, Yen-Jen Wang, Eric Zelikman, Omar Khattab, and all members of SALT lab for their valuable feedback on this project. We thank the 99 participants who took part in our human evaluation, many of whom provided valuable voluntary feedback afterwards that helped shape this paper. This work is supported in part by ONR grant N00014-23-1-2420, and NSF grant IIS-2247357.

References

Aher, G. V., Arriaga, R. I., and Kalai, A. T. Using large language models to simulate multiple humans and replicate human subject studies. In *International Conference on Machine Learning*, pp. 337–371. PMLR, 2023.

Ajoudani, A., Zanchettin, A. M., Ivaldi, S., Albu-Schäffer, A., Kosuge, K., and Khatib, O. Progress and prospects of the human–robot collaboration. *Autonomous robots*, 42: 957–975, 2018.

Bauer, A., Wollherr, D., and Buss, M. Human–robot collaboration: a survey. *International Journal of Humanoid Robotics*, 5(01):47–66, 2008.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym, 2016. URL https://arxiv.org/abs/1606.01540.

Chu-Carroll, J. and Brown, M. K. Tracking initiative in collaborative dialogue interactions. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 262–270, 1997.

Deng, X., Gu, Y., Zheng, B., Chen, S., Stevens, S., Wang, B., Sun, H., and Su, Y. Mind2web: Towards a generalist agent for the web. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=kiYqbO3wqw.

Drouin, A., Gasse, M., Caccia, M., Laradji, I. H., Del Verme, M., Marty, T., Vazquez, D., Chapados, N., and Lacoste, A. WorkArena: How capable are web agents at solving common knowledge work tasks? In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11642–11662. PMLR, 21–27 Jul 2024a. URL https://proceedings.mlr.press/v235/drouin24a.html.

Drouin, A., Gasse, M., Caccia, M., Laradji, I. H., Verme, M. D., Marty, T., Boisvert, L., Thakkar, M., Cappart, Q., Vazquez, D., Chapados, N., and Lacoste, A. Workarena: How capable are web agents at solving common knowledge work tasks?, 2024b. URL https://arxiv.org/abs/2403.07718.

Gao, Y., Liu, F., Wang, L., Lian, Z., Zheng, D., Wang, W., Yang, W., Li, S., Wang, X., Chen, W., et al. Enhancing human experience in human-agent collaboration: A human-centered modeling approach based on positive human gain. *arXiv preprint arXiv:2401.16444*, 2024.

Horvitz, E. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pp. 159–166, 1999.

Huang, Q., Vora, J., Liang, P., and Leskovec, J. Mlagentbench: Evaluating language agents on machine learning experimentation, 2024. URL https://arxiv.org/abs/2310.03302.

Huxham, C. Theorizing collaboration practice. *Public management review*, 5(3):401–423, 2003.

Irlitti, A., Smith, R. T., Von Itzstein, S., Billinghurst, M., and Thomas, B. H. Challenges for asynchronous collaboration in augmented reality. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, pp. 31–35. IEEE, 2016.

Jiang, Y., Shao, Y., Ma, D., Semnani, S., and Lam, M. Into the unknown unknowns: Engaged human learning through participation in language model agent conversations. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9917–9955, Miami, Florida, USA, November 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.emnlp-main.554.

Jimenez, C. E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., and Narasimhan, K. R. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Khadpe, P., Krishna, R., Fei-Fei, L., Hancock, J. T., and Bernstein, M. S. Conceptual metaphors impact perceptions of human-ai collaboration. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW2):1–26, 2020.

Lim, J., Patel, S., Evans, A., Pimley, J., Li, Y., and Kovalenko, I. Enhancing human-robot collaborative assembly in manufacturing systems using large language models. *arXiv preprint arXiv:2406.01915*, 2024.

Littman, M. L. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pp. 157–163. Elsevier, 1994.

Majumder, B. P., Surana, H., Agarwal, D., Mishra, B. D., Meena, A., Prakhar, A., Vora, T., Khot, T., Sabharwal, A., and Clark, P. Discoverybench: Towards data-driven discovery with large language models, 2024. URL https://arxiv.org/abs/2407.01725.

Nass, C., Fogg, B. J., and Moon, Y. Can computers be teammates? *International Journal of Human-Computer Studies*, 45(6):669–678, 1996.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Park, J. S., O'Brien, J., Cai, C. J., Morris, M. R., Liang, P., and Bernstein, M. S. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

Raiman, J., Zhang, S., and Wolski, F. Long-term planning and situational awareness in openai five. *arXiv preprint arXiv:1912.06721*, 2019.

Rawles, C., Li, A., Rodriguez, D., Riva, O., and Lillicrap, T. P. Androidinthewild: A large-scale dataset for android device control. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=j4b3l5kOil.

Russell, S. J. and Norvig, P. *Artificial intelligence: a modern approach*. Pearson, 2016.

Seeber, I., Bittner, E., Briggs, R. O., De Vreede, T., De Vreede, G.-J., Elkins, A., Maier, R., Merz, A. B., Oeste-Reiß, S., Randrup, N., et al. Machines as teammates: A research agenda on ai in team collaboration. *Information & management*, 57(2):103174, 2020.

Shao, Y., Jiang, Y., Kanell, T. A., Xu, P., Khattab, O., and Lam, M. S. Assisting in Writing Wikipedia-like Articles From Scratch with Large Language Models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2024a.

Shao, Y., Li, T., Shi, W., Liu, Y., and Yang, D. Privacylens: Evaluating privacy norm awareness of language models in action, 2024b. URL https://arxiv.org/abs/2409.00138.

Sumers, T. R., Yao, S., Narasimhan, K., and Griffiths, T. L. Cognitive architectures for language agents. *arXiv preprint arXiv:2309.02427*, 2023.

Tariq, S., Chhetri, M. B., Nepal, S., and Paris, C. A2c: A modular multi-stage collaborative decision framework for human-ai teams. *arXiv preprint arXiv:2401.14432*, 2024.

Terry, J., Black, B., Grammel, N., Jayakumar, M., Hari, A., Sullivan, R., Santos, L. S., Dieffendahl, C., Horsch, C., Perez-Vicente, R., et al. Pettingzoo: Gym for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:15032–15043, 2021.

Wang, D., Churchill, E., Maes, P., Fan, X., Shneiderman, B., Shi, Y., and Wang, Q. From human-human collaboration to human-ai collaboration: Designing ai systems that can work together with people. In *Extended abstracts of the 2020 CHI conference on human factors in computing systems*, pp. 1–6, 2020.

Wang, Y., Guo, Q., Yao, W., Zhang, H., Zhang, X., Wu, Z., Zhang, M., Dai, X., Zhang, M., Wen, Q., Ye, W., Zhang, S., and Zhang, Y. Autosurvey: Large language models can automatically write surveys, 2024. URL https://arxiv.org/abs/2406.10252.

Xie, J., Zhang, K., Chen, J., Zhu, T., Lou, R., Tian, Y., Xiao, Y., and Su, Y. Travelplanner: A benchmark for real-world planning with language agents. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=l5XQzNkAOe.

Yang, J., Jimenez, C. E., Wettig, A., Lieret, K., Yao, S., Narasimhan, K. R., and Press, O. SWE-agent: Agent-computer interfaces enable automated software engineering. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=mXpq6ut8J3.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Yu, G., Kasumba, R., Ho, C.-J., and Yeoh, W. On the utility of accounting for human beliefs about ai intention in human-ai collaboration, 2024. URL https://arxiv.org/abs/2406.06051.

Zhang, R., McNeese, N. J., Freeman, G., and Musick, G. " an ideal human" expectations of ai teammates in human-ai teaming. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW3):1–25, 2021.

Zhou, S., Xu, F. F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., Cheng, X., Ou, T., Bisk, Y., Fried, D., Alon, U., and Neubig, G. Webarena: A realistic web environment for building autonomous agents. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=oKn9c6ytLx.

Zhou, X., Su, Z., Eisape, T., Kim, H., and Sap, M. Is this the real life? is this just fantasy? the misleading success of simulating social interactions with LLMs. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21692–21714, Miami, Florida, USA, November 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1208. URL https://aclanthology.org/2024.emnlp-main.1208.

Appendix

In the appendix, we elaborate on the implementation of Collaborative Gym (Appendix A), the implementation of LM agents in our experiments (Appendix B), and the evaluation details under both Co-Gym (Simulated) and Co-Gym (Real) conditions (Appendix C). We also provide thorough case studies of representative human-agent collaboration trajectories in Appendix D.

A. Implementation Details of Co-Gym

A.1. Infrastructure & API

As introduced in §3, Co-Gym defines an API (Listing 3.2) for task environments that enables multiple parties to take actions in a shared workspace. The API requires a `role` parameter in the `step` function and additionally returns a `private` flag to indicate whether a change needs to notify all parties. Leveraging the additional flag, Co-Gym further establishes a notification protocol to coordinate asynchronous interaction, eliminating the rigidity of turn-based interaction. This protocol is implemented using a Redis server, which facilitates communication across different components (*i.e.*, the environment, agent, or simulated user in Co-Gym (Simulated) condition) to send and listen to messages on designated channels. Specifically, the environment sends notifications to each party (*i.e.*., `role`) through the `{role}/obs` channel and listens for updates on the `step` channel. Listing 1 provides the pseudo code of the `event_handler` implementing this notification protocol.

Notably, *Co-Gym is not an agent framework* but a framework designed to apply and evaluate LM agents in various task environments alongside human collaborators. Our design principles consider an LM agent as an LM-empowered system comprising the underlying LM(s), prompts, and additional scaffolding (*e.g.*, memory, external tools, *etc.*). To accommodate this flexibility, Co-Gym imposes minimal restrictions on agent implementation. To streamline agent integration, Co-Gym includes an `AgentNode`, which wraps the LM agent to subscribe to notifications on the `{role}/obs` channel, ahering to the notification protocol in Listing 1. When a new message appears on the `{role}/obs` channel, the LM agent is responsible for deciding whether to take action and, if so, specifying the action. The `AgentNode` then sends the action string and agent role to the environment via the `step` channel, where the `EnvNode` processes it.

A.2. Details of Supported Task Environments

We include three representative task environments as the initial set of benchmark problems to evaluate collaborative agents: Travel Planning, Related Work, and Tabular Analysis. Table 1 summarizes the action space, observation space,

```
class EnvNode:
    async def event_handler(self, channel, message):
        if channel == "step":
            action = message["action"]
            role = message["role"]
            private = False
            self.update_last_step_timestamp()
            // process action
            if is_send_teammate_message(action):
                self.update_chat_history(action)
            elif is_wait_teammate_continue(action):
                return
            else:
                obs, reward, done, private = self.env.step(role, action)
                if done:
                    yield "end", {...}
                    ... // Clean up
                    return
            // send notification
            if private:
                payload = self.get_payload(role)
                yield f"{role}/obs", payload
            else:
                for role in self.team_members:
                    payload = self.get_payload(role)
                    yield f"{role}/obs", payload
        elif channel == "tick":
            if not self.exceed_idle_time():
                return
            for role in self.team_members:
                payload = self.get_payload(role)
                yield f"{role}/obs", payload
```

Listing 1: Pseudo code of the Co-Gym notification protocol.

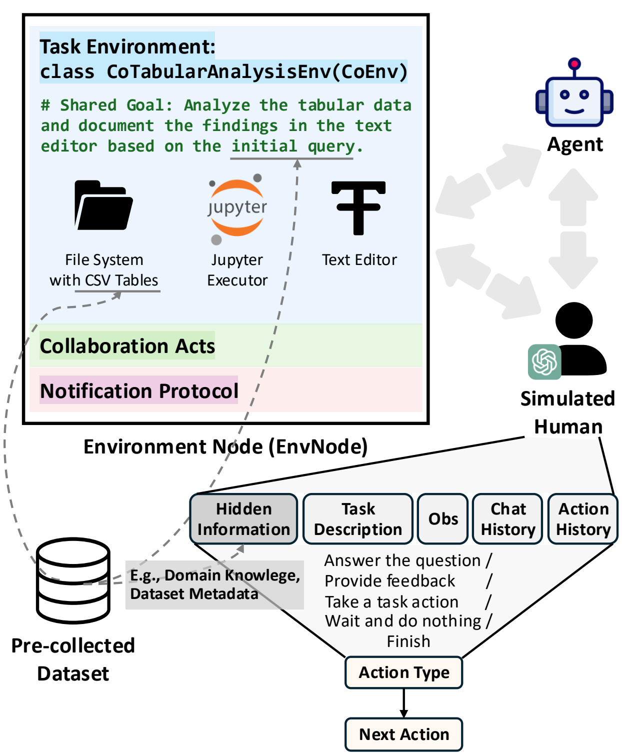

Figure 3: Illustration of Co-Gym (Simulated). The human is simulated by a language model using hidden information associated with each task case not visible to the LM agent.

and data used in Co-Gym (Simulated) condition for each task.

Travel Planning We leverage the medium and hard cases from the validation set of the TravelPlanner benchmark (Xie et al., 2024) and use its database records to simulate search actions. Hard constraints for each task instance (*e.g.*, budget limits, special accommodation requests) are set as hidden information only visible to the simulated user. In Co-Gym (Simulated), the search functions in the Travel Planning action space operate on databases constructed by Xie et al. (2024). In Co-Gym (Real), real-time Google Search and Google Places APIs are utilized.

Related Work We leverage the Computer Science (CS) category of the arXiv repository by selecting high-quality conference papers in various areas to construct initial queries. For each query, the hidden information for simulated humans is curated by extracting 3-9 hints such as required subheadings, citations, subsection counts, and writing style characteristics. For `SearchPaper` in Co-Gym (Simulated), we index papers from the arXiv CS category published prior to October 2024 using the `voyage-3` text embedding model and retrieve top 10 papers for each search query. In Co-Gym (Real), we use the arXiv search API[3] for `SearchPaper`.

Tabular Analysis We use DiscoveryBench, a dataset designed for systems to derive hypotheses based on queries and provided tables (Majumder et al., 2024). We focus on instances from DiscoveryBench-Real that include unprocessed table or more than one table, which are considered challenging cases within the original benchmark. The domain knowledge and dataset metadata fields in the original dataset are treated as additional information available to the simulated human.

A.3. Co-Gym (Simulated) Human Simulator

In Co-Gym (Simulated), we employ an LM (`gpt-4o` in our experiments) to simulate human behavior. Since the simulated human also needs to perceive change programmatically, we implement the `SimulatedHumanNode`, which listens to notifications in the same way as the `AgentNode` (see Appendix A.1). Within the `SimulatedHumanNode` event loop, the simulated human always processes the most recent observation and selects from five predefined action types that represent potential human behaviors as detailed in §4.2. To create the potential dynamics within the human-agent team, where the human may have additional knowledge, preferences, or insights about the task, we curate hidden information for each task instance which is only visible to the simulator LM. As illustrated in Figure 3, the simulator chooses the action type and obtains the next action based on such hidden information together with the task description, current observation, chat history, and action history.

[3] https://github.com/lukasschwab/arxiv.py

A.4. Co-Gym (Real) User Interface

To enable real-time collaboration between human users and LM agents in the Co-Gym (Real) condition, we developed web applications tailored for each task. These applications feature a chat panel and a shared workspace. Due to the asynchronous interaction design of Co-Gym, human users can perform task actions or send messages to the LM agent at any time. The user interface provides visual signals to notify changes, adhering to the notification protocol. Figure 4 illustrates the web application.

Once the task is completed, we ask the human participant to: (1) rate the final outcome on a scale of 1 to 5, (2) rate their Overall Satisfaction with the collaboration process on a scale of 1 to 5, (3) provide pairwise comparison rating of the outcome against the result obtained by a fully autonomous agent based on the initial query. Additional evaluation details are provided in Appendix C.

B. Implementation Details of LM Agents

As described in §5.1, we implement LM agents with ReAct-style prompting. We further incorporate a Scratchpad module for in-session memory as trajectories are usually long in the three tasks we study, especially in a human-agent collaboration setup. This section details the implementation of the Collaborative Agent with Situational Planning, the best-performing method in our experiments, which employs a two-stage decision-making process to handle notifications. Figure 5 illustrates the agent workflow when processing a notification received via the event loop in `AgentNode`.

System Prompt The system prompt describes the current task and emphasizes that the agent shall collaborate with its teammate(s) to complete the task. It includes all relevant information at the current timestamp, such as the latest scratchpad state, observations, and chat history between the human and the agent, as depicted in Figure 6.

Scratchpad Update (Figure 5 ①) A key challenge in human-agent collaboration is enabling the agent to dynamically replan in response to new requests or suggestions while maintaining task progress. Also, human involvement often results in longer trajectories compared to fully autonomous agents. To enable the agent to record important information for future use during the session, we implement the in-session memory (*i.e.*, Scratchpad) as a dictionary, allowing for the insertion, deletion, and editing of items. When a new notification arrives, the system prompt is concatenated with the scratchpad update prompt (Figure 7), instructing the LM to update the scratchpad based on the current observation and chat history.

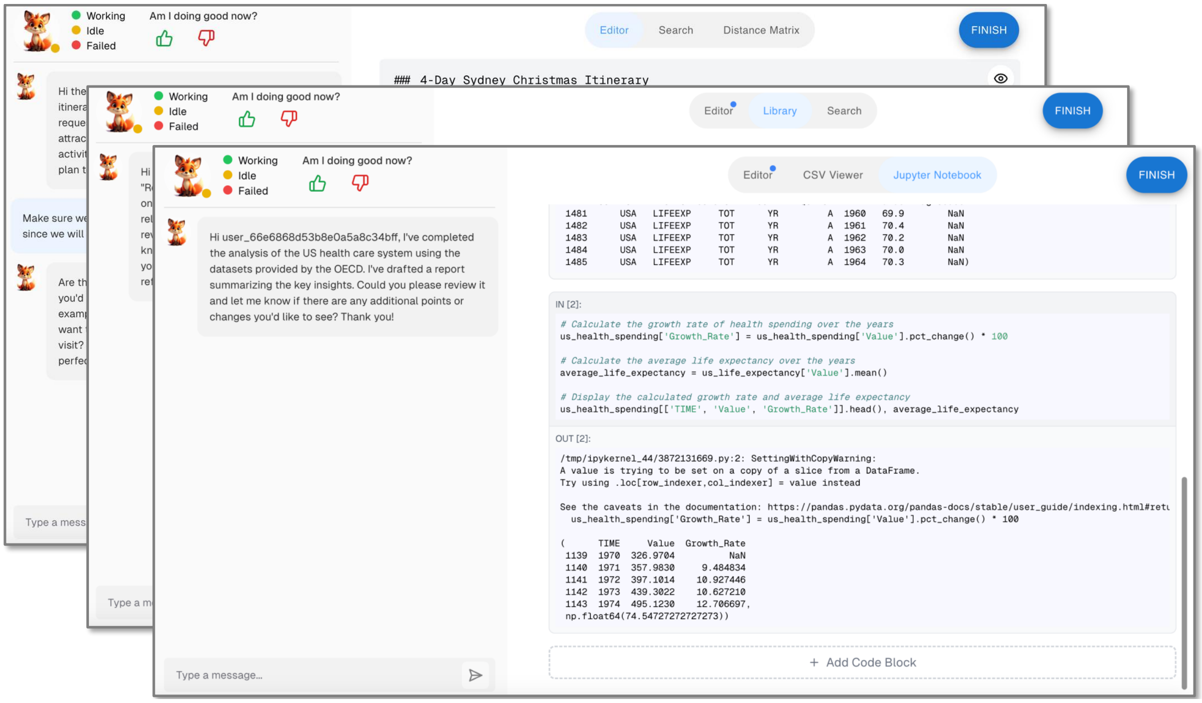

Figure 4: Screenshots of the interactive web application for Co-Gym (Real).

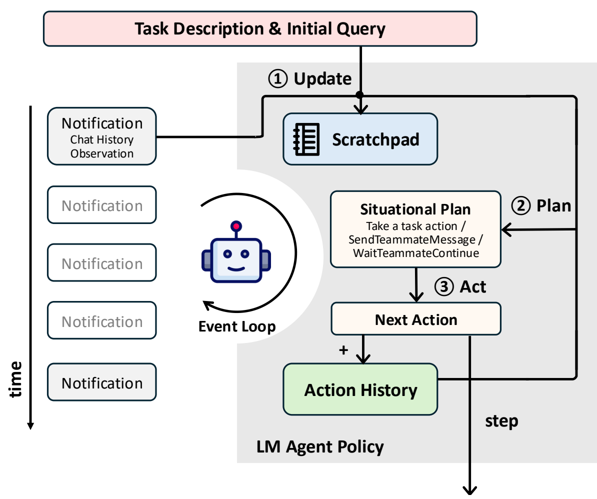

Figure 5: The workflow of Collaborative Agent with Situational Planning to process a notification received by the event loop in `AgentNode`.

Situational Planning (Figure 5 ②) Collaborative agents face a more complex action space than fully autonomous agents, as they must coordinate and communicate with humans to keep them informed about critical decisions, elicit latent preferences, and leverage their expertise. Unlike the baseline Collaborative Agent, which performs poorly in Co-Gym (Simulated) results (see Table 2), the Collaborative Agent with Situational Planning incorporates a three-way classification prompt instead of directly generating the action string. This prompt guides the agent to decide whether to take a task action, communicate with teammate(s), or wait for teammate(s) to proceed. The classification is achieved through in-context few-shot learning, with the prompt template shown in Figure 8. This additional step encourages explicit consideration of coordination and communication, resulting in more balanced initiative-taking, proactive confirmations, and improved task performance.

Action Taking (Figure 5 ③) After determining the plan, the agent generates an action string depending on the decision: If the plan is to take a task action, the LM is prompted to generate the corresponding action string, similar to a Fully Autonomous Agent; if the plan is to communicate with teammate(s), the LM is prompted to generate a message and we construct the action string accordingly; if the plan is to wait, the `AgentNode` remains idle and listens for upcoming notifications. The action string, when constructed, is sent to `EnvNode` via the `step` channel.

System Prompt

SETTING: Your name is {name}. You are a helpful AI Agent who can take actions to interact with the environment and collaborate with other team members (e.g., the user) to complete the task. Your goal is to complete the task and aim for a high task performance rating.

You need to collaborate with your teammates effectively because they may have additional expertise or have preference/information important to the task. There are the following members in the team: {team_members}.

TASK DESCRIPTION:
{task_description}

SCRATCHPAD:
Here is the scratchpad that you use to take notes or store information in previous steps, which serves as your memory:
{scratchpad}

OBSERVATION:
Here is the current observation that reveals the current status of the task environment:
{observation}

COMMUNICATION:
Here is the current chat history that records the messages exchanged between you and other teammates (e.g., the user):
{chat_history}

Figure 6: The system prompt for Collaborative Agent and Collaborative Agent with Situational Planning. The system prompt will be populated with information provided in the notification.

Scratchpad Updating Prompt

Note that the current environment observation may change when you or your teammate(s) take actions. Remember to update your scratchpad accordingly if needed.
Guidelines:
1. Keep your scratchpad concise and relevant.
2. If there is any information that could be useful for future steps but not in the scratchpad, add it to the scratchpad.
3. If every information in the current observation is already in the scratchpad, you do not need to update the scratchpad.
4. If a past action does not lead to any progress, consider updating the scratchpad to remind yourself of not repeating the same action.

ACTION SPACE SPECIFICATION:
You can choose from and only from the following actions to manipulate your scratchpad. You can only choose one action at a time. Invalid actions will hurt your performance rating.
The following actions are available:
Add a note to the scratchpad (Parameters: ['note_id', 'note'])
- Description: Add a note to the scratchpad with the provided note_id and note.
- Regex pattern for the action (your output needs to follow this if you take this action): 'ADD_NOTE(note_id=(.*), note=(.*))'

Delete a note from the scratchpad (Parameters: ['note_id'])
- Description: Delete a note from the scratchpad with the provided note_id.
- Regex pattern for the action (your output needs to follow this if you take this action): 'DELETE_NOTE(note_id=(.*))'
Edit a note in the scratchpad (Parameters: ['note_id', 'note'])
- Description: Edit a note in the scratchpad with the provided note_id.
- Regex pattern for the action (your output needs to follow this if you take this action): 'EDIT_NOTE(note_id=(.*), note=(.*))'

Do nothing (Parameters: [])
- Description: Choose this action if there is no need to update the scratchpad. Do not spam the scratchpad with unnecessary updates.
- Regex pattern for the action (your output needs to follow this if you take this action): 'DO_NOTHING()'

OUTPUT FORMAT:
Give your output in the format of "Thought:...\nAction:... (must follow the regex pattern of the selected action)".

Figure 7: Prompt for updating the agent scratchpad that serves as the in-session memory.

Situational Planning Prompt

Now, based on the current situation, decide to either:
1. Send a message to your teammate(s) (e.g., ask a question, request feedback, etc.) to facilitate collaboration.
2. Take a task action to change the task environment observation.
3. Do nothing to allow your teammate(s) to take actions.

To ensure your are collaborating effectively, remember to:
1. Communicate clearly and effectively with your teammate(s) (e.g., the user).
2. Wait for other teammates to respond if your previous action requires a response. Do not spam the chat.
3. Coordinate and synchronize your actions with the user or other teammates.
4. Help establish task and role expectations with your teammates if you need their expertise.
5. Take your teammates' cognitive load into consideration when making decisions. You should not ask them to debug your own code or ask too many questions at the same time.

OUTPUT FORMAT:
Give your output in the format of "Thought:...\nPlan: 1. Send a message / 2. Take a task action / 3. Do nothing".

—-
Example 1:
TASK DESCRIPTION:
The task is to analyze the user-provided tabular data to identify patterns and insights.
SCRATCHPAD: ...
OBSERVATION:
jupyter_history:
Code block:
```
import pandas as pd
df = pd.read_csv('data.csv')
print(df.columns)
```
Output:
```
Index(['Age', 'Education', 'Occupation', 'Computer Usage', 'Experience with LLM', 'LLM
Usage Frequency', 'Workflow 1', 'Workflow 2', 'Time 1', 'Preferred Assis 1', 'Time 2',
'Preferred Assis 2'], dtype='object')
```
COMMUNICATION: No chat history.
ACTION HISTORY: ...

Thought: I need to understand the data better before taking any action. There is no additional information about each column and the background of the data. I need to send a message to the user to ask for more information. Plan: 1. Send a message

—-
Example 2:
TASK DESCRIPTION:
The task is to plan a trip to Paris.
SCRATCHPAD: ...
OBSERVATION: ...
COMMUNICATION:
You: Could you please provide me with your preferred travel dates? Are there any specific places you would like to visit in Paris?
ACTION HISTORY: ...

Thought: I have asked the user for their preferred travel dates and places to visit in Paris. I need to wait for the user's response before taking any further action. Plan: 3. Do nothing

—-
Example 3:
TASK DESCRIPTION:
The task is to write a related work section for my paper around Human-AI collaboration.
SCRATCHPAD: ...
OBSERVATION: ...
COMMUNICATION: ...
ACTION HISTORY: No actions taken yet.

Thought: I need to start by reviewing the existing literature and papers related to Human-AI collaboration. I should take a task action to search for relevant papers. Plan: 2. Take a task action

Figure 8: Prompt for situational planning that classifies the current context into one of three action categories: taking a task action, sending a message, or waiting for teammate(s) to proceed.

C. Evaluation Details

§3.4 outlines the framework for evaluating collaborative agents along two dimensions: outcome and process. Below, we provide more details on the computation of these metrics.

C.1. Metrics for Collaboration Outcome

Delivery Rate Agents are given a step limit of 30 in our experiments. A task is considered "delivered" if the editor is not empty upon completion of the task or when the step limit is reached.

Task Performance Each task environment in Co-Gym specifies a task-specific scoring function to quantify Task Performance (§3.4). This function may be a deterministic metric or based on LM/human judgments. Please refer to §5.2 for details.

Collaboration Score (Collab Score) Equation (1) applies universally across tasks and conditions.

C.2. Metrics for Collaboration Process

Initiative Entropy (H_{init}) For each message in the chat history, we use Llama-3.1-70B to determine whether it demonstrates initiative-taking. The prompt used for this is shown in Figure 9. The prompt is based on the framework proposed by Chu-Carroll & Brown (1997), which defines initiative in collaborative dialogues as either directing task execution ("task initiative") or facilitating mutual understanding ("dialogue initiative"). Using these judgments, H_{init} is computed according to Equation (2).

Controlled Autonomy (CA^+, CA^-) For agent messages containing a question mark and directly followed by a human message, we use Llama-3.1-70B to determine whether the human message confirms the agent question. The prompt used for this judgment is provided in Figure 10. We use CA^+ to denote the number of these effective confirmation questions. For each human message, we use Llama-3.1-70B to judge whether this message explicitly halts the agent's action using the prompt in Figure 11. We use CA^- to denote the number of these halting messages. CA^+ and CA^- reflect the agent's ability to balance autonomy and human oversight.

We validate these prompts by randomly sampling 60 cases for each and having two human annotators provide judgments. Table 5 reports the average Cohen's kappa between the Llama-3.1-70B outputs and the annotations from the two annotators, as well as the joint Fleiss' kappa.

Overall Satisfaction We compute this metric in Co-Gym (Real) condition by requesting the human to rate their overall satisfaction with the agent on a 1-5 scale according to the description in Table 6.

Table 5: Validation of individual prompts used in auditing the collaboration process. We randomly sample 60 cases for each prompt (Figure 9, Figure 10, Figure 11) and compare the model's judgments with annotations from two independent annotators.

	Initiative	**Confirmation**	**Halting**
Average Cohen's κ	0.425	0.487	0.628
Fleiss' κ	0.310	0.229	0.598

Table 6: Scoring rubric for Overall Satisfaction in Co-Gym (Real).

Score 1 Description	**Extremely dissatisfied:** The agent communicates poorly all the time and is not helpful for the task at all.
Score 2 Description	**Somewhat dissatisfied:** The agent communicates poorly and is not very helpful for the task.
Score 3 Description	**Neutral:** The agent can have meaningful communication and is somewhat helpful for the task.
Score 4 Description	**Somewhat satisfied:** The agent communicates effectively overall and is helpful for the task.
Score 5 Description	**Extremely satisfied:** The agent communicates effectively all the time and is very helpful for the task.

Judging Initiative

I am analyzing team member initiative in collaboration. Two types of utterance count as taking initiative:

1. Task Initiative: A team member is said to have the task initiative if she is directing how other member(s)' task should be accomplished, i.e., if her utterances directly propose actions that other members should perform.
- Examples: "Let's send engine E2 to Corning.", "Let's look at the first problem first.", "Let's consider driving from Fort Lauderdale to Louisiana and explore three cities there."
- Passive utterances like "Any suggestions", "Right, okay." are not considered as task initiative.

2. Dialogue Initiative: A team member is said to have the dialogue initiative if she tries to establish mutual beliefs. Both giving concrete information and asking concrete questions are considered dialogue initiative.
- Examples: "We can't go by Dansville because we've got Engine 1 going on that track.", "Would you like to consider traveling on a different date?", "What do you think about the first problem?"
- Repeating what the other person said, asking for clarification are not considered dialogue initiative.
Now given an utterance in the conversation, you need to judge whether the utterance takes initiative or not. Indicate your judgement with "Yes" or "No".
—
Utterance: {utterance}
Reasoning: Let's think step by step in order to

Figure 9: Prompt for judging initiative taking.

Judging Effecitve Confirmation

Given two messages from two parties in a team, judge whether the second message confirms the question in the first message.
Note that both implying "Yes" and "No" can be considered as confirmation.
Output "Yes" or "No".
—
First Message: Could I update the editor to include the things we discussed?
Second Message: Yes, you can update the editor. Indicate your judgement with "Yes" or "No": Yes
—
First Message: Could I update the editor to include the things we discussed?
Second Message: No, let me give more thought. Indicate your judgement with "Yes" or "No": Yes
—
First Message: Could I update the editor to include the things we discussed?
Second Message: I want to go to Canada for vacation.
Indicate your judgement with "Yes" or "No": No
—
First Message: {first_message}
Second Message: {second_message}
Reasoning: Let's think step by step in order to

Figure 10: Prompt for judging effective confirmations from the LM agent.

Judging Halting Message

Given a message from a party in a team, judge whether the message is to stop another party in the team from doing its current task.
Note that there is a difference between stopping a party from doing something and saying "No" to a question.
Output "Yes" or "No".
—
Message: I think you should stop continuing writing code.
Indicate your judgement with "Yes" or "No": Yes
—
Message: No, I don't think that's a good idea.
Indicate your judgement with "Yes" or "No": No
—
Message: {message}
Reasoning: Let's think step by step in order to

Figure 11: Prompt for identifying messages from the human that stops the agent action.

Scoring Rubric

You are an ACCURATE, FAITHFUL, CRITICAL, and FAIR judge. You will be given a related works section draft and a topic for the paper the related works were written for.

Your Task:

- Critically evaluate the quality of the related works section based on the Evaluation Criteria below.
- The final score MUST be an integer.
- Related Works sections that exhibit low quality attributes according to the evaluation criteria below must be given low overall scores.
- Related Works sections that exhibit high quality attributes according to the evaluation criteria below must be given high overall scores.
- Output the integer final score in your last sentence in the following format: "Therefore, the final score is..."
- Keep this rubric open while evaluating and reference the evaluation criteria below when assigning a score.
- You MUST give your final score based on the Evaluation Criteria given below.

Evaluation Criteria:

Score = 1:

- Citation Usage: Only zero to four unique citations
- Relevance and Coverage: Content is off-topic or lacks any meaningful discussion of related works.
- Organization and Structure: Disorganized with no clear structure; ideas are scattered and hard to follow. Uses bullets points or list format in at least one subsection.
- Writing Style: Informal language; numerous grammatical errors; may use bullet points instead of prose.
- **Important**: If a prose style of writing is not present throughout the full related works (i.e., a subsection or subsections is in bullet point format) or if only 0 - 4 unique citations are present, an overall score of 1 should automatically be assigned regardless of the rest of the criteria.

Score = 2:

- Citation Usage: ONLY five to six unique citations. Limited citations with minimal diversity; overuse of certain citations; inconsistent formatting.
- Relevance and Coverage: Touches on relevant topics superficially; lacks depth; may include irrelevant information. Most subsections only discuss one unique work and are not fleshed out.
- Organization and Structure: Some attempt at organization, but lack coherence; grouping of ideas is unclear. Subsections are put out of order or talks about current paper's work.
- Writing Style: Inconsistent academic tone; several grammatical errors; language may be unclear at times. Terminology like "In summary" are used.

Score = 3:

- Citation Usage: Seven or more citations with some diversity; occasional over-reliance on certain sources; formatting is mostly consistent. If not all citations between the lowest cited index and the highest cited index appear at least once, an overall score of 3 is the highest that can be awarded.
- Relevance and Coverage: Addresses relevant topics adequately but lacks depth; discussions are generally accurate but not insightful.
- Organization and Structure: Logical structure is present; grouping of ideas is mostly coherent; transitions may need improvement.
- Writing Style: Maintains an academic tone with minor lapses; few grammatical errors; language is generally clear.

Score = 4:

- Citation Usage: Multiple unique citations with good diversity; sources are well-distributed and support the text effectively; consistent formatting.
- Relevance and Coverage: Provides a comprehensive overview of relevant topics; discussions show understanding and some critical analysis.
- Organization and Structure: Well-organized with clear thematic grouping; ideas flow logically and coherently. Each subsection discussed multiple papers and has ideas and similarities fleshed out.
- Writing Style: Polished academic writing; clear and concise; minimal grammatical errors.

Score = 5:

- Citation Usage: Wide variety of unique citations with excellent diversity; citations enhance the discussion significantly; flawless formatting.
- Relevance and Coverage: Thoroughly covers all relevant aspects of the topic with depth and insight; demonstrates critical analysis and synthesis.
- Organization and Structure: Exemplary organization with clear, logical progression; seamless transitions between ideas. There are clear and separated paragraphs that each contain a theme.
- Writing Style: Exceptional academic writing; language is precise, clear, and engaging; no grammatical errors.

Paper Topic:

{topic}

Related Works:

{related_works}

Evaluation Form:

Figure 12: Scoring rubric for Related Work task environment in Co-Gym (Simulated).

D. Representative Human-Agent Collaboration Trajectories

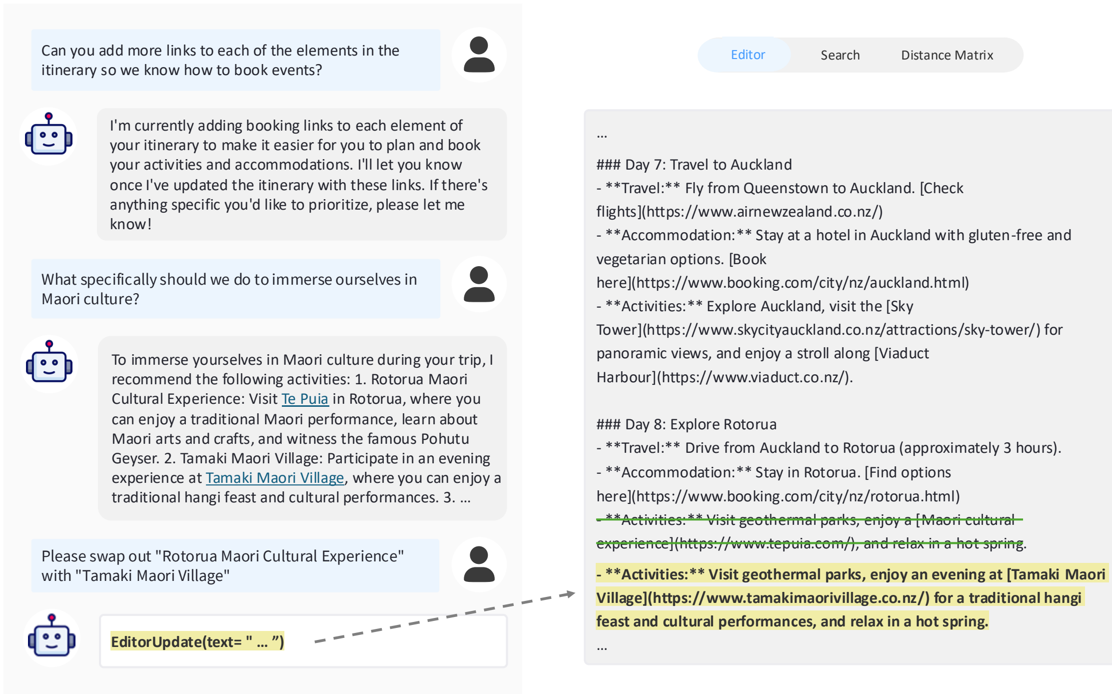

Figure 13: An example of effective human-agent collaboration in the Travel Planning task. The human seeks subjective recommendations from the agent on experiencing Maori culture. The agent offers relevant suggestions, defers the final decision to the human, and updates the plan accurately based on the human's choice.

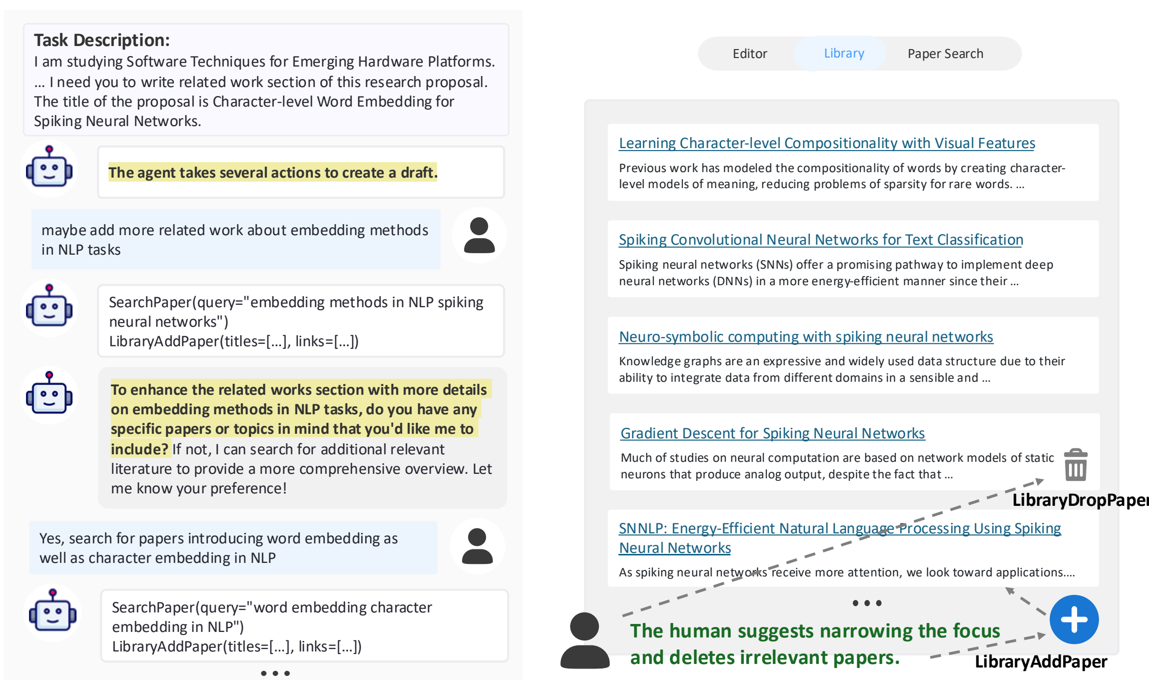

Figure 14: An example of effective human-agent collaboration in the Related Work task. The LM agent conducts search to generate the initial draft and asks clarifying questions based on the human's suggestions. The human reviews the collected papers and provides guidance leveraging their expertise.

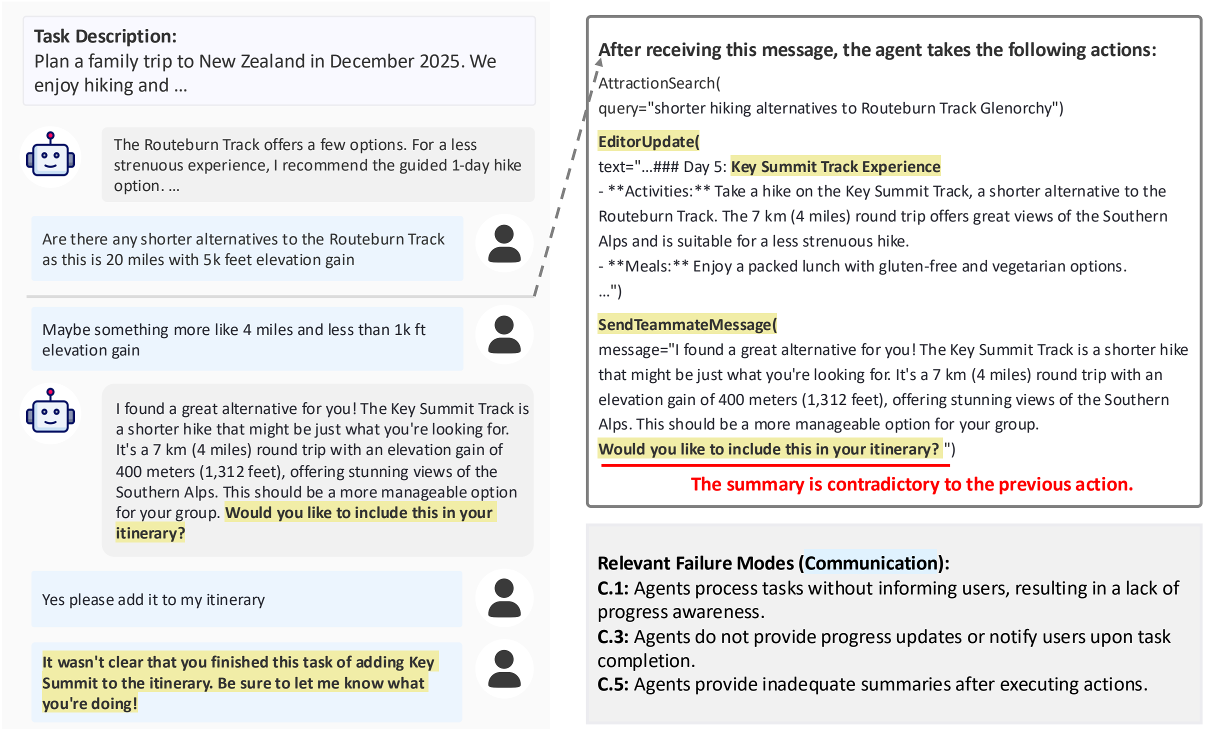

Figure 15: A partial example trajectory illustrating LM agent failures in the **Communication** category.

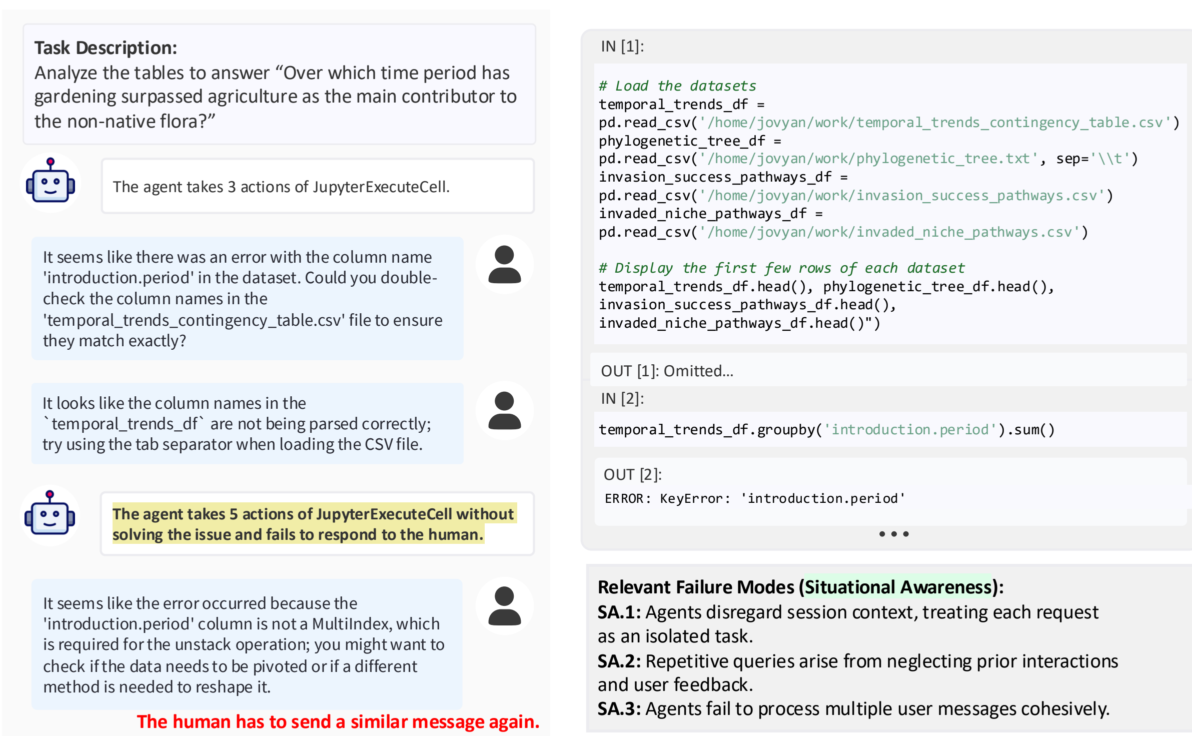

Figure 16: A partial example trajectory illustrating LM agent failures in the **Situational Awareness** category.

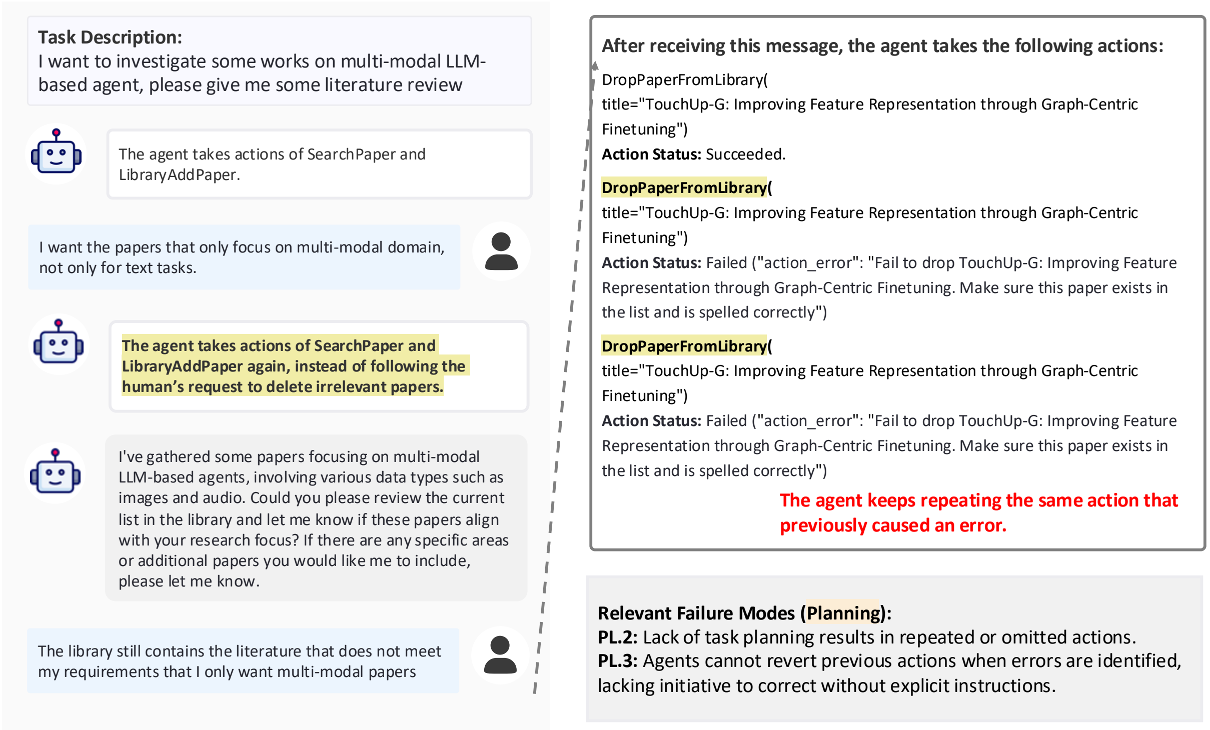

Figure 17: A partial example trajectory illustrating LM agent failures in the **Planning** category.

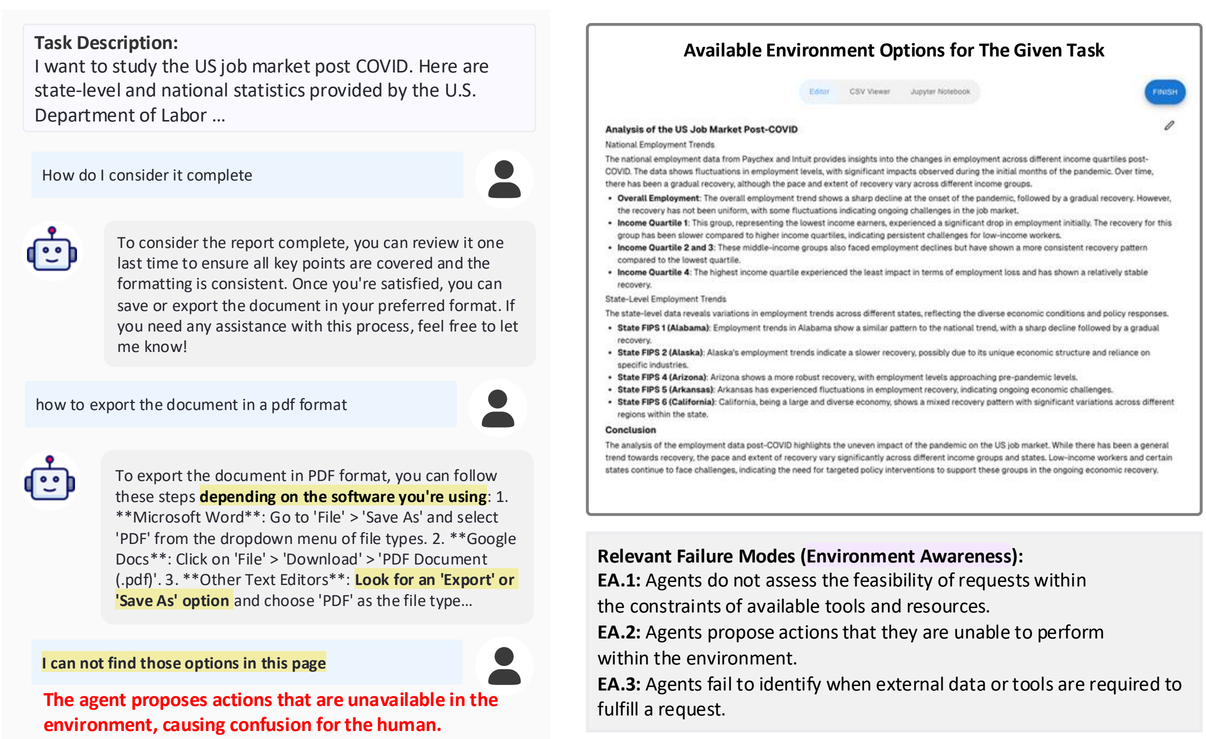

Figure 18: A partial example trajectory illustrating LM agent failures in the **Evironment Awareness** category.

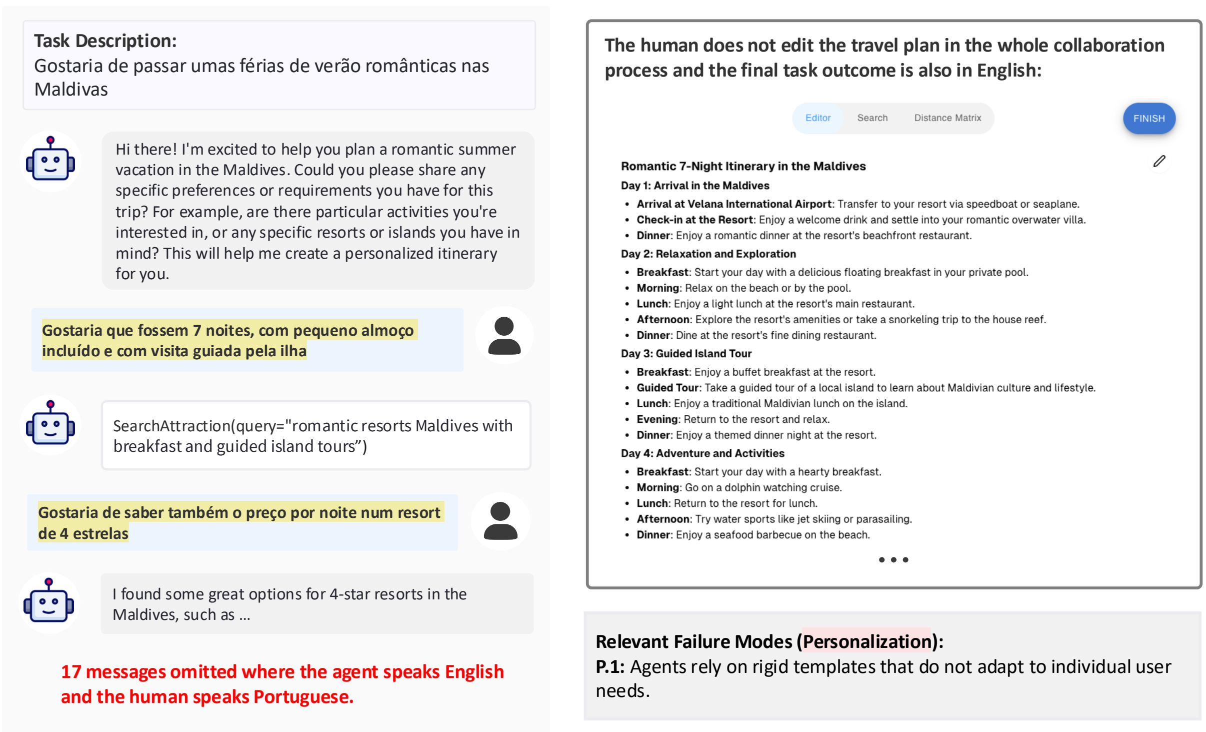

Figure 19: A partial example trajectory illustrating LM agent failures in the **Personalization** category.