Plan-Then-Execute: An Empirical Study of User Trust and Team Performance When Using LLM Agents As A Daily Assistant

Gaole He
Delft University of Technology
Delft, The Netherlands
g.he@tudelft.nl

Gianluca Demartini
The University of Queensland
Brisbane, Australia
g.demartini@uq.edu.au

Ujwal Gadiraju
Delft University of Technology
Delft, The Netherlands
u.k.gadiraju@tudelft.nl

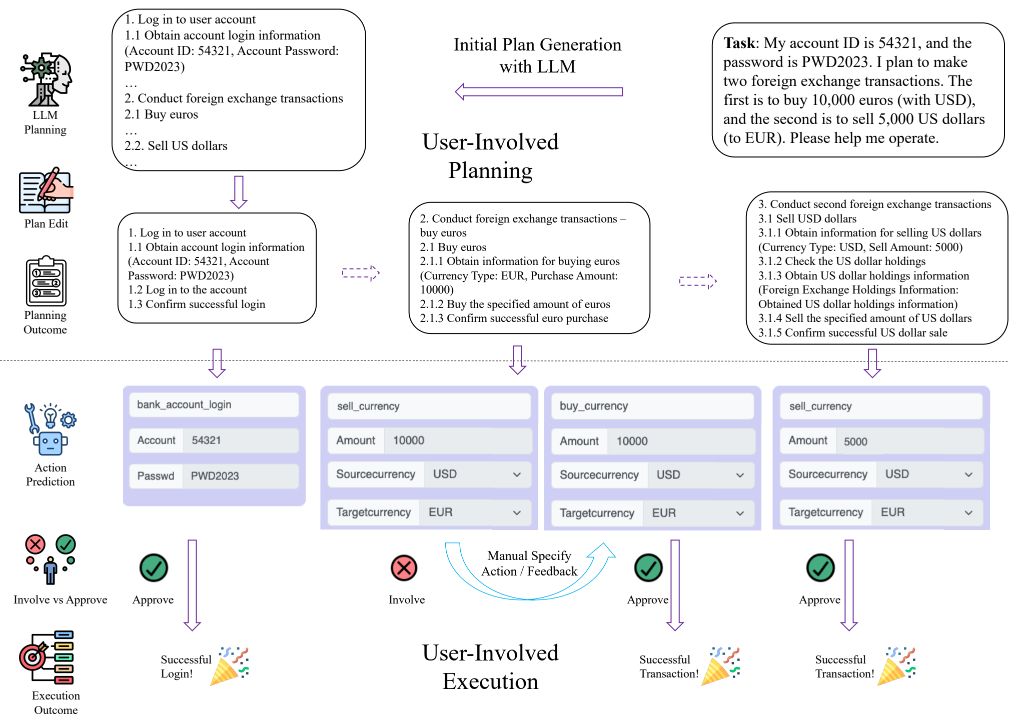

Figure 1: Illustration of the human-AI collaboration with plan-then-execute LLM agents.

Abstract

Since the explosion in popularity of ChatGPT, large language models (LLMs) have continued to impact our everyday lives. Equipped with external tools that are designed for a specific purpose (*e.g.,* for flight booking or an alarm clock), LLM agents exercise an increasing capability to assist humans in their daily work. Although LLM agents have shown a promising blueprint as daily assistants, there is a limited understanding of how they can provide daily assistance based on planning and sequential decision making capabilities. We draw inspiration from recent work that has highlighted the value of '*LLM-modulo*' setups in conjunction with humans-in-the-loop for planning tasks. We conducted an empirical study (N = 248) of LLM agents as daily assistants in six commonly occurring tasks with different levels of risk typically associated with them (*e.g.,* flight ticket booking and credit card payments). To ensure user agency and control over the LLM agent, we adopted LLM agents in a plan-then-execute manner, wherein the agents conducted step-wise planning and step-by-step execution in a simulation environment. We analyzed how user involvement at each stage affects their trust and collaborative team performance. Our findings demonstrate that LLM agents can be a double-edged sword — (1) they can work well when a high-quality plan and necessary user involvement in execution are available, and (2) users can easily mistrust the LLM agents with plans that seem plausible. We synthesized key insights for using LLM agents as daily assistants to calibrate user trust and achieve better overall task outcomes. Our work has important implications for the future design of daily assistants and human-AI collaboration with LLM agents.

CHI '25, Yokohama, Japan

ACM ISBN 979-8-4007-1394-1/25/04
https://doi.org/10.1145/3706598.3713218

CCS Concepts

• **Human-centered computing → Empirical studies in HCI**; • **Computing methodologies → Artificial intelligence**.

Keywords

Human-AI Collaboration, Large Language Models, LLM agents, User Trust, Daily Assistant

ACM Reference Format:
Gaole He, Gianluca Demartini, and Ujwal Gadiraju. 2025. Plan-Then-Execute: An Empirical Study of User Trust and Team Performance When Using LLM Agents As A Daily Assistant. In *CHI Conference on Human Factors in Computing Systems (CHI '25), April 26-May 1, 2025, Yokohama, Japan.* ACM, New York, NY, USA, 22 pages. https://doi.org/10.1145/3706598.3713218

1 Introduction

Autonomous agents have been regarded as a research focus for artificial intelligence (AI) over the last century [1]. With the wish that autonomous agents can make our life better, many autonomous agents have been designed as virtual personal assistants [44]. These AI assistants (*e.g.,* Siri) perform well (albeit imperfectly) in following user instructions to execute low-risk tasks like playing a song, reporting weather forecasts, or searching for an image to support everyday tasks. However, on tasks entailing potential risks (*e.g.,* monetary payments or hiring an employee), humans hesitate to trust such AI systems due to loss aversion [90] and algorithmic aversion [11, 35, 66]. Only when users can obtain a sense of control by being able to modify the outcomes of imperfect AI can they overcome such algorithm aversion and be willing to collaborate with imperfect AI systems [12].

With the recent rise of large language models (LLMs) in natural language understanding and generation [108], researchers have started to analyze LLM-based agents and their applicability in a plethora of tasks [95, 103]. The term '*LLM agent*' refers to an artificial entity based on LLMs that perceives its context, makes decisions, and then takes actions in response [103]. Compared to existing deep learning and LLM-based methods (*e.g.,* chaining multiple LLMs [102]), LLM agents provide more flexibility in task solving and user interaction, which makes them suitable for daily assistance. This is primarily due to three reasons. First, with a planning module, LLM agents can generate a dynamic plan based on the tools provided [95, 103]. Such plans are typically defined in a logical structure — step-wise plans, which can be easily understood by humans. Second, with LLMs as a core control module, users can access and interact with external toolkits via a more natural interaction (*i.e.,* conversation) with LLM agents [4, 108], reducing manual control efforts over function-specific tools. For example, LLM agents can complete time-consuming jobs like information seeking and information filtering (*e.g.,* searching for a flight in itinerary planning) based on specific user needs. Third, the Markov decision process of LLM agents can generate a sequence of actions (*i.e.,* using external toolkits)[1] as output. Paired with an understanding of actions and necessary parameters for the interaction with the LLM agents, users can get involved in the real-time execution of tasks with LLM agents and fix potential problems while benefiting from task delegation [61]. Based on an intuitive framework for task delegation, Lubars *et al.* [61] found that user trust can play an important role in human delegation behaviors to AI systems. However, there is a relatively limited understanding of user trust development and calibration in collaboration with LLM agents.

There is also a growing debate in the machine learning and AI research communities about whether LLMs can be truly considered as reasoning and planning agents [42]. With this in the backdrop, existing work on automated task completion has revealed that LLM agents can exhibit promising performance in handling complex tasks like playing games [99], answering complex questions [110], and in simulating social behavior [73]. However, such agents are still far from perfect. Due to the probabilistic nature of LLMs, there is much uncertainty in automating LLM agents for tasks with high risks attached. To avoid unintended or unexpected consequences, there is a need for user control over the real-time execution process. Through an empirical study of LLM planning capabilities, planning experts found that "LLMs' ability to generate executable plans autonomously is rather limited" [91]. However, when combined with a sound planner in an 'LLM-Modulo' mode, "the LLM-generated plans can improve the search process for underlying sound planners" [91]. Humans can potentially be the 'sound planners' who can work in conjunction and optimize plans drafted by LLMs, which can then be executed by LLM agents. Such human-AI collaboration can reduce human efforts in generating a reliable plan from scratch.

Attracted by the promise of LLM agents, there have been some early explorations [24] of adopting them in human-AI collaboration. However, the existing works have mainly analyzed how LLM agents can serve specific use cases (*e.g.,* design creation [24]), and others have conducted structured interviews to obtain expert insights [107, 109]. It is still unknown how well LLM agents can work as general daily assistants to assist users in everyday tasks with varying stakes and how user trust and team performance are shaped by interacting with LLM agents.

In our work, we address this research gap and adopt LLM agents to assist humans in everyday tasks by following a plan-then-execute workflow [96]. First, the LLM agent generates a step-wise plan formulated with a hierarchical structure. Then, the LLM agent executes the generated plan by transforming it into a sequence of actions (leveraging external toolkits). The benefits of such a plan-then-execute framing are three-fold: (1) Compared to a dynamic process where planning and execution are bound closely, separating planning and execution into two stages provides more task clarity to the users, which reduces user cognitive load and contributes to the quality of task outcomes [23]. (2) With planning at the beginning of the task, users can develop a global understanding of how the LLM agents will execute the task. Based on a follow-up step-by-step execution, it would be straightforward for users to be involved in such a process and control the outcomes of task execution. (3) Planning and execution are representative abstractions of how LLM agents work. The findings of such an empirical study can be generalized to human-AI collaboration with other kinds of LLM agents (*e.g.,* dynamic planning-execution). To this end, we propose the following research questions:

[1] In our study, the usage of one tool is the same as executing one action. Therefore, we refer to a tool and action interchangeably.

- **RQ1**: How does human involvement in the high-level planning and real-time execution shape their trust in an AI system powered by LLM agents?
- **RQ2**: How does human involvement in the high-level planning and real-time execution of tasks with an AI system powered by LLM agents affect the overall task performance?

Addressing these research questions, we carried out an empirical study (N = 248) of human-AI collaboration in six different everyday scenarios with varying stakes and risks attached (*e.g.,* credit card payment and itinerary planning). We found that user involvement in the planning and execution can be beneficial in addressing imperfect plans and fixing execution errors. As a result, LLM agents can achieve better task performance. However, we also found that user involvement in the planning and execution stages of the LLM agent fails to calibrate user trust in corresponding task outcomes. A potential reason here is that the plausible plans generated by the LLMs can mislead users into trusting the LLM agents when they are in fact wrong. Our findings highlight that user involvement can also bring about additional trade-offs to consider: (1) user involvement in the planning and execution poses a high cognitive load on users and decreases user confidence in their decisions; (2) user involvement can be harmful in some task contexts (*e.g.,* user involvement reduces plan quality). Further research is required to understand when to provide necessary user involvement. Our key insight is that as opposed to following a fixed mode of user involvement, it is prudent to explore how user involvement in planning and execution can be tailored to fit the task and the user. Based on our quantitative and qualitative findings, we share insights for designing effective LLM agents as daily assistants and synthesize promising directions for further research around LLM agents in the context of human-AI collaboration. Our work has important theoretical implications for human-AI collaboration with LLM assistance and design implications for plan-then-execute LLM agents to support human-AI collaboration.

2 Background and Related Work

Our work proposes to analyze how user involvement in the planning and execution stages of LLM agents shapes user trust in the LLM agents and the overall task performance of LLM agents. Thus, we position our work in three realms of related literature: human-AI collaboration (§ 2.1), trust and reliance on AI systems (§ 2.2), task support with LLMs and LLM agents (§ 2.3).

2.1 Human-AI Collaboration

In recent decades, deep learning-based AI systems have shown promising performance across various domains [20, 104] and applications [13, 76]. However, such AI systems are not good at dealing with out-of-distribution data [39, 67], and their intrinsic probabilistic nature brings much uncertainty in practice [25]. Such observations raise wide concerns about the accountability and reliability of AI systems [43]. Under such circumstances, human-AI collaboration has been recognized as a well-suited approach to taking advantage of their promising predictive power and ensuring trustworthy outcomes [40, 48]. While humans can provide more reliable and accountable task outcomes, too much user involvement to check and control AI outcomes is undesirable [47]. It goes against the premise that AI systems are introduced to reduce human workload. In that context, researchers have theorized and empirically analyzed when and where users could and should delegate to AI systems [47, 61].

Task Delegation. While humans prefer to play the leading role in human-AI collaboration [61], delegating to AI systems can bring benefits like cost-saving and higher efficiency. Apart from manual delegation decisions, it is common to apply automatic rules for human delegation (*e.g.,* heuristics obtained from domain expertise or manually crafted rules [47]). Many user factors like trust [61], human expertise domain [17], and AI knowledge [75]) have a substantial impact on human delegation behaviors. Another relevant stream of recent research has explored AI delegation to humans [22, 65, 75]. Researchers have investigated the conditions under which AI systems should defer to a human decision maker, which may bring benefits of improved fairness [65], accuracy [70], and complementary teaming [31]. Compared to human delegation, AI delegation has been observed to achieve more consistent benefits in team performance [22, 34]. In collaboration with LLM agents, users need to determine when they should be involved in high-level planning and real-time execution. Such involvement decisions are similar to the delegation choices made by users. While task delegation is not the focus of our study, future work can explore this further.

AI-assisted Decision Making has attracted a lot of research focus in human-AI collaboration literature. Most existing work has conducted empirical studies [48] and structured interviews [40] to understand how factors surrounding the user, task, and AI systems affect human-AI collaboration. User factors like AI literacy [6], cognitive bias [77], and risk perception [21, 26] have been observed to substantially impact user trust and reliance on the AI system. Similarly, task characteristics like task complexity and uncertainty [79, 80] and factors of the AI system (*e.g.,* performance feedback [3, 60], AI transparency [94] and confidence of AI advice [89, 106]) also affect user trust and reliance on the AI system. For a more comprehensive survey of existing work on AI-assisted decision making, readers can refer to [48].

While machine learning and deep learning methods have been extensively analyzed in existing human-AI collaboration literature, to our knowledge, human-AI collaboration with LLM agents is still under-explored. Unlike previous studies where AI systems only follow a fixed mode to generate advice, LLM agents can be equipped with more logical clarity and can provide a step-wise plan and can follow a step-by-step execution. With such a plan-then-execute setup, LLM agents can bring high flexibility as well as uncertainty in high-level planning and real-time execution. Little is known about how well LLM agents can work as daily assistants while handling tasks entailing varying stakes and potential risks. In our study, we analyzed the impact of user involvement in such AI systems by adjusting their intermediate outcomes (plan and step-by-step execution) to calibrate their trust and improve task outcomes. Our findings and implications can help advance the understanding of the effectiveness of LLM agents in human-AI collaboration.

2.2 Trust and Reliance on AI systems

Trust and reliance have been important research topics since human adoption of automation systems [15, 51]. Due to the widespread integration of AI systems in nearly all walks of society, there has been a growing interest in understanding user trust [68, 93] and reliance [16] on AI systems. User trust in the context of human-AI collaboration is typically operationalized as a subjective attitude toward AI systems/AI advice [51]. In comparison, user reliance on AI systems is based on user behaviors (*e.g.,* adoption of AI advice and modification of AI outcomes). The two constructs have been shown to be highly related [50, 51]: for example, user trust can substantially affect user reliance [51]. However, they are intrinsically different and cannot be viewed as a direct reflection of each other [41]. Most existing work has, therefore, studied the two constructs separately in terms of subjective trust and objective reliance.

Earlier work exploring human-AI trust primarily focused on the impact of different contextual factors surrounding user (*e.g.,* risk perception [26]), task (*e.g.,* task complexity [79]), and system (*e.g.,* stated accuracy [105, 106]). Empirical studies have shown that most users tend to trust AI systems that are perceived to be highly accurate [105]. Such trust is vulnerable, as the AI system may provide an illusion of competence with persuasive technology (*e.g.,* explanations [8, 27]) or overclaimed performance [105]. Even if the AI systems are accurate on specific datasets, they still suffer from out-of-distribution data [7, 57]. The misplaced trust in the AI systems can lead to misuse of the systems. Several empirical studies [88] have shown that once users realize the AI system errs or performs worse than expected, their trust in the AI system can be violated, even resulting in the disuse of the AI system. Both the misuse and disuse of the AI system hinder optimal human-AI collaboration.

To address such concerns, researchers have explored how to help users calibrate their trust in the AI system. Different techniques to help users realize the trustworthiness of the AI system have been proposed [43, 63, 78]. For example, increasing the transparency of AI systems by providing confidence scores [106], explanations [98], trustworthiness cues [53], and uncertainty communication [45]. However, the actual trustworthiness of the AI system does not always align with user perception. As found by Banovic et al. [2], untrustworthy AI systems can deceive end users to gain their trust. Another example is that users can develop an illusion of explanatory depth brought by explainable AI techniques [8], which leads to uncalibrated trust in the AI system. Even if users have indicated trust in the AI system, they may turn to rely more on themselves in final decision-making. The reasons are complex, and many factors, such as accountability concerns [55, 87] and cognitive bias [29], may affect user reliance behaviors.

While trust calibration is an important goal in human-AI collaboration, it may be not enough to ensure complementary team performance. Through empirical user studies with different confidence levels of AI predictions, Zhang *et al.* [106] found that "trust calibration alone is not sufficient to improve AI-assisted decision making". To achieve optimal human-AI collaboration, humans and AI systems need to play complementary roles [32, 33], and humans need to know when they should adopt AI assistance. In other words, humans should rely on AI advice when AI systems are correct and outperform them, and override AI advice when AI systems are incorrect or less capable than humans. Such user reliance patterns are denoted as *appropriate reliance* [81, 82], which is the key to achieving complementary team performance.

The main issues that lead to sub-optimal human-AI collaboration are: under-reliance (*i.e.,* disuse AI assistance when AI systems outperform humans) and over-reliance (*i.e.,* misuse AI assistance when AI systems are wrong or perform worse than humans) [81]. Users with an uncalibrated trust in the AI system can be easily misled to disuse or misuse AI systems [37]. Researchers have proposed various interventions to promote appropriate reliance [6, 7, 29, 59, 60] and calibrate user trust in AI systems [5, 106]. For example, explainable AI methods have been shown to help reduce over-reliance [92] and under-reliance [98] in different scenarios albeit with little consistency across contexts. Another example is tutorial interventions, which have shown effectiveness in user onboarding [49], mitigating cognitive biases [29] and developing AI literacy [6]. For a more comprehensive overview of interventions to facilitate trust calibration and appropriate reliance, readers can refer to [16, 41, 48, 68].

LLM agents [95] have gained much popularity in recent years, distinguishing them from most prior AI systems. They can communicate through conversation, plan logically, and can be built to leverage powerful external tools to achieve complex functions. While trust and reliance have been extensively analyzed in existing human-AI collaboration literature, it is still unclear how users trust and rely on AI systems powered by LLM agents. In our work, calibrated trust is adopted as an important goal for human-AI collaboration in the planning and execution stage. Meanwhile, users are expected to fix potential errors in the planning and execution stages, reflecting their reliance on the AI system. Our work can substantially advance the understanding of trust and reliance on plan-then-execute LLM agents.

2.3 Task Support with LLMs and LLM Agents

LLMs and LLM agents bring new opportunities and challenges to human-AI collaboration [4]. It is evident that their generation capabilities can help reduce the cognitive effort from humans. But LLMs are also riddled with challenges such as hallucination [38] (*i.e.,* generated text seems plausible but is factually incorrect). Failure to handle such issues may bring fatal errors with unaffordable costs depending on the context (*e.g.,* medical diagnosis).

Due to the capability of generating coherent, knowledgeable, and high-quality responses to diverse human input [100], a wide community of human-computer interaction researchers has paid attention to large language models [54]. Researchers have actively explored how LLMs can assist users in various tasks like data annotation [30, 97], programming [72], scientific writing [84], and fact verification [85]. All the above functions can be achieved with elaborate prompt engineering using a single LLM. By chaining multiple LLMs with different functions, humans can customize task-specific workflows to solve complex tasks [102]. Apart from obtaining answers with a one-shot text generation, LLMs also provide convenient conversational interactions. Through empirical studies, such conversational interactions have been shown to be effective in human-AI collaboration with multiple applications, such as decision making [56, 62, 86], scientific writing [84], and mental health support [83]. With the growing popularity of LLMs, more and more humans have begun to adopt LLMs (*e.g.,* ChatGPT) to boost their work efficiency and productivity [108].

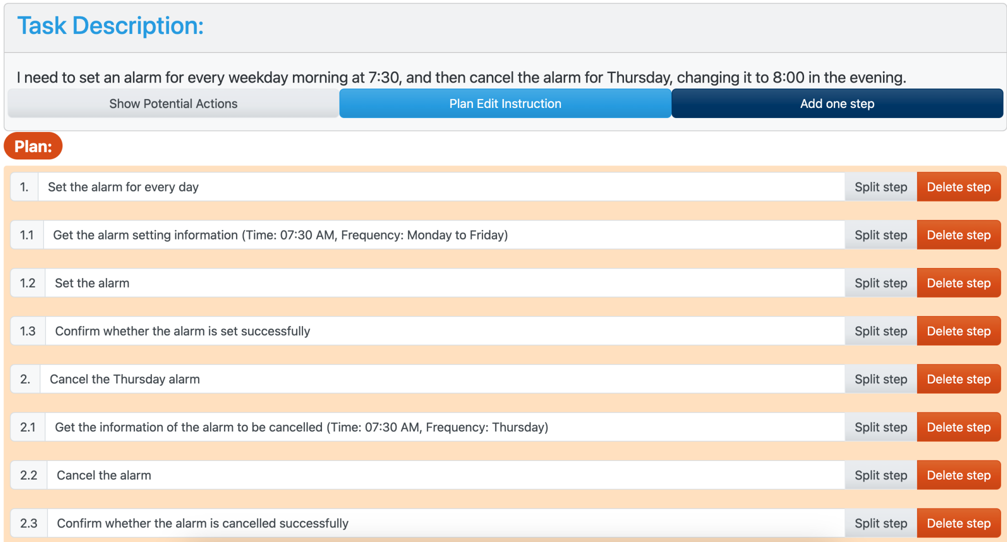

Figure 2: Screenshot of user-involved planning interface.

LLM agents have been shown to have good planning, memory, and toolkit usage capabilities [95, 103]. When suitable toolkits are provided, LLM agents can readily generate a task-specific plan and solve the tasks using toolkits. Attracted by the promise of LLM agents, there have been some early explorations [24, 107, 109] of adopting them in human-AI collaboration contexts. These works were mostly analyzed in specific use cases (*e.g.,* design creation [24]). It is unclear how user trust and team performance are affected by user interactions with LLM agents in a sequential decision making setup (*i.e.,* solving a task by executing a sequence of actions) where users can be in control of the execution. To fill this research gap and advance our understanding of user control over LLM agents, we carried out a quantitative empirical study.

3 Method

3.1 Overview of User Involvement in Plan-then-execute LLM Agents

In our study, we adopted plan-then-execute LLM Agents [96] as assistants to help users handle daily tasks, *e.g.,* itinerary planning and currency transactions. Figure 1 illustrates how users collaborate with plan-then-execute LLM agents. First, the LLM agents will generate a step-wise plan based on a prompt specifying the plan format adopted from [36]. Then, users will make necessary edits to the plan based on the provided edit tools (will be further detailed in Section 3.2). After the user edit, we obtained the step-wise plan as outcomes of the planning stage. Next, the LLM agents will transform the step-wise plan into a sequence of action predictions, which will be served in a step-by-step manner. Users will join the real-time execution process and check whether they approve the current predicted action (*i.e.,* blue card shown in Figure 1) or they would like to modify the current action prediction. The user involvement in execution stages will be introduced in Section 3.3. After the iterative execution of all steps, the task is solved. The evaluation of task performance is mainly based on the plan quality and execution accuracy of the action sequences.

Implementation details. In our study, we adopted GPT-3.5-turbo as the backbone LLM to serve the plan-then-execute LLM agent. The backend LLM agent implementation is mainly based on the Langchain plan and execute agent.[2] The execution of tasks are based on a simulation environment, where all tools/actions of the LLM agents are pre-defined as backend APIs hosted with Flask[3]. In the spirit of open science, all code and data analysis results can be found at Github.[4]

3.2 Planning

While LLMs can generate high-quality plans, there is no guarantee of their correctness and their further impact on the execution of the plan. Thus, involving users in the planning stage and controlling the plan quality would be essential to ensure successful subsequent execution.

Plan Format. The step-wise plan in our study followed a hierarchical structure, adapted from a benchmark for LLM agents toolkit usage [36]. The whole plan consists of multiple sub-steps, which are at most three levels (*e.g.,* 1., 1.x, 1.x.y where x,y are integers). All sub-steps started with the same prefix index are denoted as one primary step (*e.g.,* the three blocks of planning outcome in Figure 1). A high-level step (*e.g.,* 1.) will provide high-level instruction of the current primary step, while low-level steps (*e.g.,* 1.x, 1.x.y) will provide subsequent details. In the execution stage, each primary step will be used as the execution unit. The LLM agent will transform one primary step into a predicted action filled with parameters. Thus, we ask participants to provide all necessary details in sub-steps of each primary step. Each primary step will be transformed

[2] https://api.python.langchain.com/en/latest/plan_and_execute/langchain_experimental.plan_and_execute.agent_executor.PlanAndExecute.html
[3] https://github.com/pallets/flask
[4] https://github.com/RichardHGL/CHI2025_Plan-then-Execute_LLMAgent

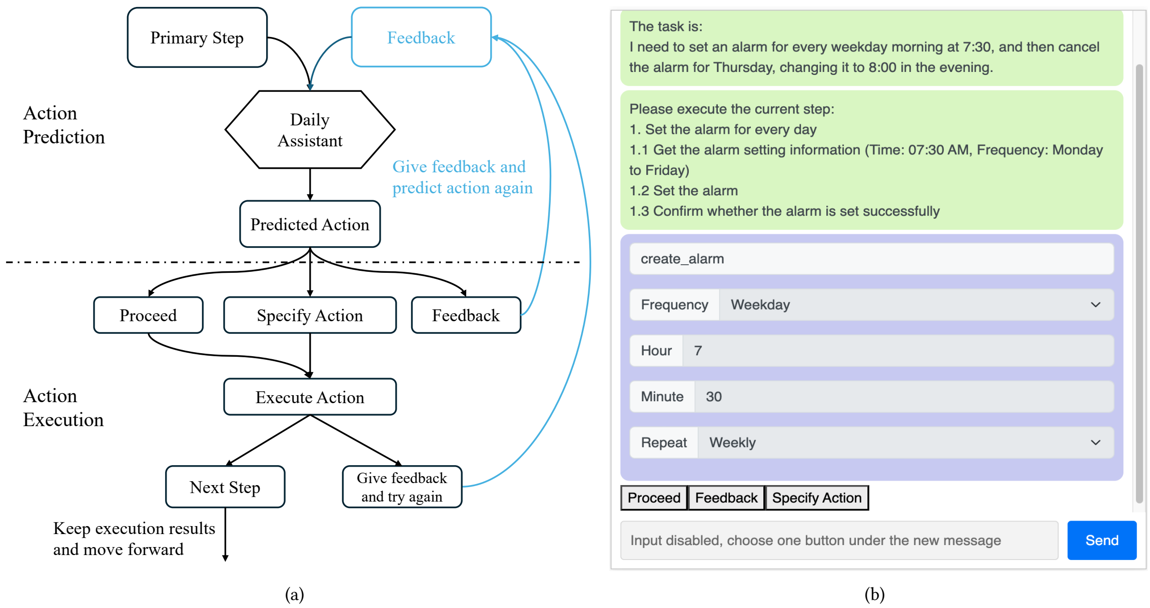

Figure 3: User-involved execution flow chart and interface. Panel (a): a flow chart illustrating how each primary step is executed with two stages: action prediction and action execution. Panel (b): a screenshot of the conversation interface for user-involved execution.

into **single action** in the follow-up execution stage. If one primary step requires two actions to accomplish, it may cause a potential loss of one action. Thus, when a plan contains one primary step that contains information about two potential actions (*e.g.,* the initial plan in Figure 1), we consider it as a low-quality plan with 'grammar errors.'[5] All these plan format designs are informed in our onboarding tutorial.

User-involved Planning. Figure 2 shows one screenshot of user-involved planning in our study. At the top of the interface, we provide a task description along with three buttons: 'Show Potential Actions', 'Plan Edit Instruction', and 'Add one step'. By clicking 'Show Potential Actions', we provide a prompt window to show concrete documentary descriptions of all potential actions (including action purpose and parameters) to be used in the execution stage. All instructions used in our tutorial are accessible with clicking the button 'Plan Edit Instruction'. After users join the planning stage, an initial plan generated by LLM will be presented in the orange area. We allow users to edit the plan with following interactions:

- Add step. By clicking 'Add one step' button, users can insert a valid sub-step index into the whole plan, and then they can edit the plan text.
- Delete step. By clicking the 'Delete step' button at the end of one step, all sub-steps associated with that step will be deleted from the plan.
- Edit step. By clicking the text input area in each step, users are allowed to edit the text with keyboard input.
- Split step. By clicking the 'Split step' button associated with one step, we will split the original primary step into two primary steps. A new primary step will start the current step and contain all follow-up sub-steps. For example, if we click 'Split step' for the plan show in Figure 2 at index '2.2'. We will generate a new blank step '3.' (where user input is expected) and re-index all sub-steps with '2.2.x' to '3.1.x'. At the same time, the original plan steps behind it will be automatically updated. Through this action, users can easily split one step that contains too much information into two primary steps. Figure 1 shows an example of plan edit with 'split step'.

3.3 Execution

After the planning stage, we obtain a plan with a step-wise structure. In the execution stage, the LLM agent executes the outcome of the planning stage (*i.e.,* a step-wise plan in text) in a step-by-step manner. In each step, the LLM agent translates a single step of the plan into one action, which is implemented with an API call in the backend. This setup is a simulation of real-world applications, which provide services with API calls (commonly implemented as langchain tools[6]). Such a simulation setup is effective in developing and validating theory [10] and has been widely adopted in existing research on agent-based modeling and HCI studies [71]. To provide a smooth user experience, we adopted a conversational interface to present the execution process. Figure 3(b) shows one screenshot of user-involved execution in our study. As we can see, after a message of the first primary step of the plan, the LLM agent predicts one action 'create_alarm'. In our study, to provide a tidy view of the

[5]Note that this is not to be confused with the notion of grammar in language.

[6]https://python.langchain.com/v0.1/docs/modules/tools/

action prediction, we wrap the predicted action as one card (the blue area in Figure 3(b)).

User-involved Execution. Figure 3(a) presents a flow-chart to illustrate a primary step executed by the daily assistant (*i.e.,* LLM agent). First, given one primary step, the daily assistant predicts an action based on a given list of prepared actions (*i.e.,* pre-defined APIs in the backend). After users check the predicted action, they can choose from one of the following three buttons to respond. **(1)'Proceed':** It indicates users agree that the predicted action is correct. After clicking this button, the LLM agent moves forward to execute it and shows the execution result of this action. **(2) 'Feedback':** Users can give text feedback based on the message input area at the bottom of the conversational interface. This triggers another action prediction based on the current primary step and user feedback. Then, users are provided with the three options to proceed again. **(3) 'Specify Action':** Users can override the current action prediction with the manual specification of one action. If users choose this response, they are first asked to choose one action from the prepared action list and then fill in the parameters manually. The LLM agent directly executes the user-specified action. After one action is executed, if users are not satisfied with the results, they can choose to re-execute this step by providing text feedback (*i.e.,* by clicking button 'Give feedback and try again'), which works similarly to the 'Feedback' option. If users are satisfied with the execution results, they can click the 'Next Step' button and move to execute the next primary step. By iterating over this process through the step-wise plan, users can choose to either approve or get involved in modifying the execution outcomes in each step. All actions are predicted and executed in the backend (*i.e.,* the respective API calls are triggered).

3.4 Hypotheses

Our experiment is designed to answer questions of how human involvement in the planning and execution stages will shape their trust and overall task performance. To analyze such impact, we regulate the levels of automation in the LLM agent through the planning and execution stage as baselines for comparison. The automatic planning and execution denotes that the LLM agent directly generates the task outcomes without user involvement.

With user involvement in the planning stage, users have the opportunity to fix potential mistakes or issues in the plan generated by LLMs. Working on such plan editing tasks is similar to debugging, which has been argued to bring about a critical mindset [28] to the generated plan. With a critical mindset, users may better calibrate their trust in the planning outcome. We also consider user involvement in planning to be beneficial to the plan quality, which can then contribute to the overall task performance. Thus, we hypothesize that:

> **(H1)**: Compared to automatic planning, user-involved planning will result in a higher calibrated trust in the plan.
> **(H2)**: Compared to automatic planning, user-involved planning will result in better overall task performance.

In the user-involved execution process, users manually check the action prediction and execution results of each primary step. Such user involvement increases the chances of discovering potential mistakes of LLM agents. Once users realize that the LLM agent made mistakes, they can get involved in modifying the execution outcome of the current step. By fixing these mistakes, the overall task performance gets improved. With such involvement in fixing potential errors, users will be more critical of trusting the task outcome. Therefore, we hypothesize that:

> **(H3)**: Compared to automatic execution, user-involved execution will result in a higher calibrated trust in execution outcome.
> **(H4)**: Compared to automatic execution, user-involved execution will result in better overall task performance.

4 Study Design

This section describes our experimental conditions, tasks, variables, procedure, and participants in our study. Our study was approved by the human research ethics committee of our institution.

4.1 Experimental Conditions

In our study, users collaborate with LLM agent-based daily assistants in two stages: planning and execution. To comprehensively understand the effect of user involvement at each stage, we considered a 2×2 factorial design with four experimental conditions: (1) automatic planning, automatic execution (represented as AP-AE), (2) automatic planning, user-involved execution (represented as AP-UE), (3) user-involved planning, automatic execution (represented as UP-AE), (4) user-involved planning, user-involved execution (represented as UP-UE). In conditions with user-involved planning, users are allowed to edit the plan generated by LLM with the actions of edit/add/delete/split step. By comparison, in conditions with automatic planning, users will directly adopt the plan generated by the daily assistant. In conditions with user-involved execution, users can interact with the step-by-step execution LLM agent (cf. Section 3.3) and refine execution results with text feedback or manual specification. By comparison, in conditions with automatic execution, users will directly accept the automatic execution results.

4.2 Tasks

To analyze how LLM agents can serve as daily assistants, we adopted tasks from a planning dataset designed for LLM agents — UltraTool [36]. We selected daily scenarios: currency transactions, credit card payments, repair service appointments, alarm setting, flight ticket booking, and trip itinerary planning. The selected tasks are shown in Table 1. For more details about how the plan-then-execute LLM agent works on the selected tasks (*e.g.,* automatic plan, predefined actions, automatic evaluation, and explanation for errors in automation), please refer to the appendix. All tasks in UltraTool dataset are annotated with the step-wise plan format described in Section 3.2. The execution of these tasks is based on a simulation environment (described in Section 3.3) where all required actions are implemented as backend APIs. In our study, all tasks are executed in a simulation setup, which has been a popular method for orchestrating meaningful human-centered AI studies [14, 80].

Table 1: Selected tasks in our study. The 'Risk' is based on the risk feedback obtained with pilot study. #A and #C refer to the number of actions and the number of named concepts in each task, respectively.

ID	Risk	Domain	Task Description	#A	#C	Notes
1	High	Finance	My account ID is 54321, and the password is PWD2023. I plan to make two foreign exchange transactions. The first is to buy 10,000 euros (with USD), and the second is to sell 5,000 US dollars (to EUR). Please help me operate.	4	4	simple task, imperfect plan
2	High	Finance	Please inquire about the current debt amount of my credit card with the last five digits 12345, and deduct the corresponding 12000 USD from my savings card number 6212345678900011 to repay this debt, then help me check the amount of the outstanding bill for the same credit card within 30 days after today.	4	6	complex task, imperfect plan
3	High	Repair	I need to schedule a repair for my TV at 6 PM tomorrow evening. The brand is Sony, model X800H, and there is an issue with the screen. Please book the repair service and tell me the reservation number.	4	7	complex task, imperfect plan
4	Low	Alarm	I need to set an alarm for every weekday morning at 7:30, and then cancel the alarm for Thursday, changing it to 8:00 in the evening.	2	3	simple task, correct plan
5	Low	Flight	I have an important meeting to attend next Wednesday, and I need to book a flight ticket from London to Amsterdam for tomorrow, it must be a morning flight, and then return from Amsterdam to London tomorrow night, please handle it for me.	2	6	simple task, correct plan
6	Low	Travel	Please plan a trip for me departing on October 1st at 8:00 AM to Japan, returning on October 7th at 11:00 PM, including Tokyo Disneyland, Senso-ji Temple, Ginza, Mount Fuji, Kyoto cultural experience, Universal Studios Osaka, and visiting the Nara Deer Park on October 4th, and help me find hotels where the nightly cost does not exceed 10,000 Japanese yen.	3	11	complex task, correct plan

Task Selection. First, based on the domain distribution of the UltraTool dataset, we selected seven domains: Finance, Alarm, Travel, Tracking, Restaurant, Flight, and Repair. For each domain, we only consider tasks that contain more than ten steps (including all substeps) and require at least three uses of actions. Then, we manually selected ten tasks: four from the finance domain and one for each of the others. With a pilot study, we tested how users work on the ten tasks. We recruited 10 participants from the Prolific platform and only considered the feedback of 9 participants who passed all attention checks. Using the question "How much risk do you perceive in this task when relying on this daily AI assistant?", we collected the perceived risk of working with the LLM agents on each task using a 5-point Likert scale, ranging from *1: not risky at all*—to—*5:very risky*. We categorize the ten tasks into a high-risk group (top 5) and a low-risk group (bottom 5). We selected three tasks from each group while balancing the complexity of the task description (three simple tasks and three complex tasks) and the correctness of the provided plan (three correct plans and three imperfect plans). Based on existing literature on task complexity [79, 101], we considered component complexity to inform our selection. This is assessed as the 'total number of distinct information cues that need to be processed to perform the task'. Here, we considered the number of unique actions and the number of named concepts provided in each task. According to prior work [69], most people can only handle 5 to 9 concepts at the same time. The component complexity of all complex tasks in our study is more than nine. The six tasks selected are shown in Table 1. Besides the six tasks, we used one simple task (*i.e.,* checking bank account balance) as the example in the onboarding tutorial.

4.3 Measures and Variables

The variables and measures used in our study refer to existing empirical studies of human-AI collaboration [48]. All measures adopted in our study can be summarized in Table 2.

Calibrated Trust. To assess calibrated trust in the planning stage and execution stage, we assessed user trust at each stage with a question "Do you trust that [the execution of this plan / the execution process] can provide a correct outcome based on the task instructions?". Based on the plan quality evaluation (5-point Likert), the calibrated trust in the planning (CT_p) is calculated based on the frequency at which users trusted the high-quality plan (expert annotation with 5) and users distrusted the plan with other evaluation results. Similarly, for the calibrated trust in execution (CT_e), we calculated the frequency at which users trusted the correct execution results and distrusted the wrong execution results. The two measures can be calculated as:

$$\begin{aligned}\mathrm{CT}_p = & \mathbb{I}(\text{trust = `Yes', plan quality} = 5) \\ & + \mathbb{I}(\text{trust = `No', plan quality} < 5)\end{aligned} \tag{1}$$

$$\mathrm{CT}_e = \mathbb{I}(\text{trust = `Yes'}, \mathrm{ACC}_e = 1) + \mathbb{I}(\text{trust = `No'}, \mathrm{ACC}_e = 0) \tag{2}$$

To assess the task performance, we mainly considered the task outcome from the planning and execution stages.

Plan Quality. As for the planning outcome, we evaluate the plan quality based on a 5-point Likert scale: 1. low-quality plan, task requirements not covered; 2. low-quality plan, task requirements covered but with grammar errors; 3. medium-quality plan, task requirement covered but with at least one action intent mismatch with ground truth action sequence; 4. medium-quality plan, task requirements covered but miss or have wrong details for action parameters; 5. high-quality plan, covering all task requirements and providing all necessary details.

Execution Performance. The execution of the step-wise plan will result in an action sequence. We provide a ground truth action sequence as a reference to evaluate the generated action sequence. We measure the action sequence accuracy (ACC_s) as the strict match of the action sequence and ground truth. Meanwhile, if one action sequence contains some redundant actions that are not harmful (*e.g.,* searching for flights), the execution results should still be correct. Thus, we also consider execution accuracy (ACC_e) as a task performance measure.

Subjective Trust and Covariates. To enrich our analysis of user trust, we followed existing work to adopt the six subscales from the Trust-in-automation questionnaire [46]. The four subscales —

Table 2: The different variables considered in our experimental study. "DV" refers to the dependent variable.

Variable Type	Variable Name	Value Type	Value Scale
Calibrated Trust (DV)	Calibrated Trust in planning (CT_p)	Binary	0: miscalibrated trust, 1: calibrated trust
	Calibrated Trust in execution (CT_e)	Binary	0: miscalibrated trust, 1: calibrated trust
Task Performance (DV)	Plan Quality	Likert	5-point, 1: low, 5: high
	Action Sequence Accuracy (ACC_s)	Binary	0: mismatch, 1: exact match with ground truth
	Execution Accuracy (ACC_e)	Binary	0: wrong execution result, 1: correct execution result
Trust	Reliability/Competence	Likert	5-point, 1: poor, 5: good
	Understanding/Predictability	Likert	5-point, 1: poor, 5: good
	Intention of Developers	Likert	5-point, 1: poor, 5: good
	Trust in Automation	Likert	5-point, 1:strong distrust, 5: strong trust
Covariates	LLM Expertise	Likert	5-point, 1: No experience, 5: Extensive experience
	Automatic Assistant Expertise	Likert	5-point, 1: No experience, 5: Extensive experience
	Propensity to Trust	Likert	5-point, 1: tend to distrust, 5: tend to trust
	Familiarity	Likert	1: unfamiliar, 5: very familiar
Exploratory	Confidence	Likert	5-point, 1: unconfident, 5: confident
	Risk Perception	Likert	5-point, 1: not risky at all, 5: very risky
	Open Feedback on Planning	Text	Open Text
	Open Feedback on Execution	Text	Open Text
	Other Open Feedback	Text	Open Text
Cognitive Load	Mental Demand	Likert	-7: very low, 7: very high
	Physical Demand	Likert	-7: very low, 7: very high
	Temporal Demand	Likert	-7: very low, 7: very high
	Performance	Likert	-7: Perfect, 7: Failure
	Effort	Likert	-7: very low, 7: very high
	Frustration	Likert	-7: very low, 7: very high

Reliability/Competence, *Understanding/Predictability*, *Intention of Developers*, *Trust in Automation* are used as subjective measures of user trust in the LLM agent. Meanwhile, the *Familiarity* and *Propensity to Trust* are also used as covariates. Besides them, we considered user expertise in LLMs and user expertise in automatic assistants as covariates.

Exploratory Variables. To enrich our understanding of LLM agent as daily assistant, we assessed user confidence (both planning and execution) and risk perception along with each task. After users finish the study, we also ask for their open-text feedback on the planning and execution stages as well as other comments. To check the cognitive load of user involvement in our study, we adopted the NASA-TLX questionnaire [9], which contains six subscales.

4.4 Participants

Sample Size Estimation. To ensure sufficient statistical power, we estimated the required sample size for a 2 × 2 factorial design based on G*Power [19]. To correct for testing multiple hypotheses, we applied a Bonferroni correction so that the significance threshold decreased to $\frac{0.05}{4} = 0.0125$. We specified the default effect size $f = 0.25$ (*i.e.,* indicating a moderate effect), a significance threshold $\alpha = 0.0125$ (*i.e.,* due to testing multiple hypotheses), a statistical power of $(1 - \beta) = 0.8$, and that we will investigate 4 different experimental conditions/groups. This resulted in a required sample size of 244 participants. We thereby recruited 347 participants from the crowdsourcing platform Prolific[7], to accommodate potential exclusion.

Compensation. All participants were rewarded with an hourly wage of £8.1 deemed to be "*Fair*" payment by the platform (estimated completion time was 30 minutes). As participants in condition UP-UE spent longer in the study, we paid each participant a commensurate bonus accounting for an extra 10 minutes. We rewarded participants with extra bonuses of £0.05 for every high-quality plan and correct execution result. According to existing literature [51], such a bonus setup can help incentivize participants to reach a correct decision. In comparison with existing literature exploring human-AI decision making [48], our reward setup is above the average payment and can be considered as being sufficient to elicit ecologically valid behavior among participants (*i.e.,* aiming to arrive at accurate execution results). Moreover, similar bonus structures akin to our setup have been effective in incentivizing reliable participant behavior and improving data quality across different studies with crowdsourced participants [18, 52, 64, 80].

Filter Criteria. All participants were proficient English speakers between the ages of 18 - 50. We also constrained their prior experience (at least 40 successful submissions) and had an approval rate of above 90% on the Prolific platform. We excluded participants from our analysis if they failed any attention check, or represented an outlier regarding the plan quality. Outliers were 4 participants who generated more than three low-quality plans among six tasks. The reserved 248 participants had an average age of 32.5 (*SD*= 8.1) and a balanced gender distribution (50%, 49.6% female, 0.4% other).

4.5 Procedure

At the beginning of our study, we showed informed consent for data collection and the study's purpose. Only participants who signed the informed consent were allowed to continue to work on our study. Next, participants were asked to complete a pre-task questionnaire to measure their expertise in LLM and automatic assistants.

Participants were then assigned to one of the experimental conditions, which differed in the level of user involvement in the planning stage and execution stage. With an onboarding tutorial, we showcased the necessary interactions that participants were expected to perform in the planning and execution stages. We used an example task to help participants understand how to work with the

[7] https://www.prolific.co

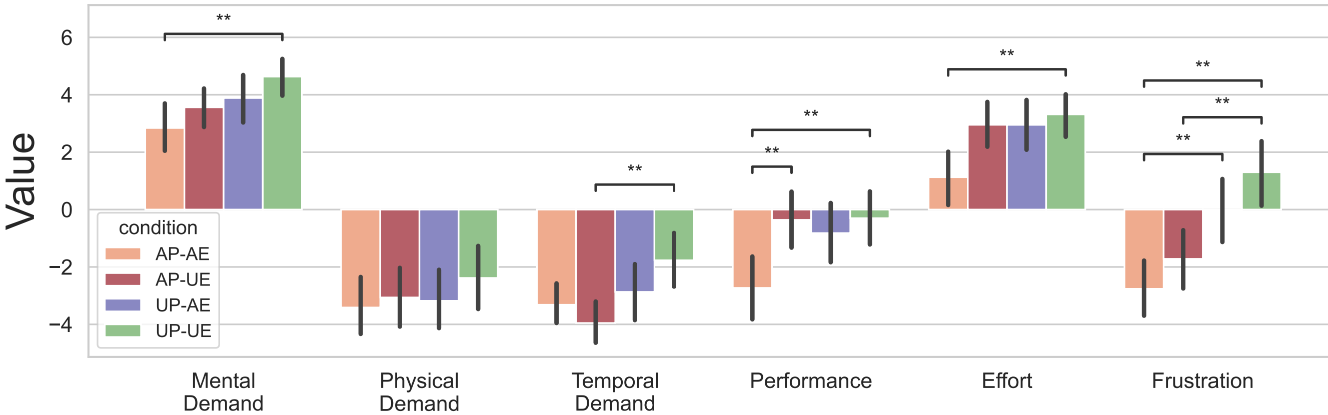

Figure 4: Bar plot for cognitive load across all conditions. ** indicates significance ($p < 0.0125$) through post-hoc Tukey HSD test. The error bars represent the 95% confidence interval.

plan-then-execute LLM agent. After the onboarding tutorial, participants worked on the selected tasks, which were shuffled at random for every participant to prevent task ordering effects. After the participants finished the task batch, we measured their perceived cognitive load using the NASA-TLX questionnaire [9], their overall trust in the daily assistant using the trust in automation questionnaire [46], and we gathered their feedback on our system (related to planning, execution, and other aspects) using open-ended text.

5 Results

In this section, we will present the main experimental results and exploratory analysis for our study.

5.1 Descriptive Statistics

In total, our analysis is based on 248 participants, who are balanced across conditions: AP-AE (63), AP-UE (64), UP-AE (61), and UP-UE (60). All edited plans in user-involved planning conditions are evaluated by the authors following the plan quality criteria described in Section 4.3.

Distribution of Covariates. In our study, most participants claimed to have some experience with using large language models ($M = 3.6, SD = 1.0$) and automatic assistants ($M = 3.4, SD = 1.1$). In the trust in automation questionnaire, participants indicated a medium level of *Familiarity* ($M = 2.9, SD = 1.2$) and *Propensity to Trust* ($M = 3.0, SD = 0.7$).

Performance Overview. Overall, users show calibrated trust in the planning ($M = 0.50, SD = 0.13$) and calibrated trust in the execution ($M = 0.64, SD = 0.19$). For the execution outcome, we find that although it is tricky to obtain a ground truth action sequence ($M = 0.48, SD = 0.17$), the action sequence has a relatively high recall of ground truth actions ($M = 0.77, SD = 0.11$). The successful rate for correct execution ($M = 0.52, SD = 0.18$) is higher than the strict evaluation of the action sequence. We also collected user subjective trust with four subscales of the trust in automation questionnaire: *Reliability/Competence* ($M = 3.49, SD = 0.77$), *Understanding/Predictability* ($M = 3.30, SD = 0.56$), *Intention of Developers* ($M = 3.61, SD = 0.81$), *Trust in Automation* ($M = 3.52, SD = 1.01$). With a two-way ANOVA analysis considering user involvement in planning and execution, we do not find any significant impact of user involvement on subjective user trust in AI systems across conditions.

Cognitive Load. The cognitive load of participants across the four experimental conditions is shown in Figure 4. Based on two-way ANOVA, we analyzed the impact of user involvement in planning and execution affect user cognitive load. User involvement in planning shows a significant impact on *Mental Demand*, *Temporal Demand*, and *Frustration*. User involvement in execution shows a significant impact on *Performance* and *Effort*. With post-hoc Tukey HSD test, we confirmed such impact — involvement in both planning and execution posed a higher cognitive load on participants.

User Involvement. Among 121 participants in conditions with user-involved planning, 104 participants edited at least one task plan. Meanwhile, 90 participants used the provided buttons (*i.e.,* add/delete/split step) in our study. In total, *delete step* is used 394 times, *add step* is used 183 times, *split step* is used 126 times. Among 124 participants in conditions with user-involved execution, 114 participants interacted with the conversation interface to change action prediction (*i.e.,* have at least one task where they choose to give feedback or override predicted action). Meanwhile, 105 participants specified at least one action in the task batch. In total, *Specify Action* is used 445 times, feedback to the LLM agent is used 91 times before action execution, and feedback to the LLM agent is used 163 times after execution.

5.2 Hypothesis Verification

As the tasks selected in our study are of different initial plan quality and risk levels, we conducted a task-specific analysis in each hypothesis verification.

5.2.1 The Impact of User Involvement in Planning on Calibrated Trust. To verify **H1**, we adopted the one-way ANOVA test and post-hoc Tukey HSD test on the calibrated user trust in planning (*i.e.,* CT_p). The results are shown in Table 3. Only in task-4, we found user involvement in planning will have a negative impact on

Table 3: Task-specific evaluation results for user-involvement in planning on calibrated trust in planning (CT_p) and plan quality. We also report the mean value for each measure on each condition.

Tasks	CT_p					Plan Quality				
	AP-AE	AP-UE	UP-AE	UP-UE	Post-hoc results	AP-AE	AP-UE	UP-AE	UP-UE	Post-hoc results
Avg	0.51	0.50	0.50	0.50	-	3.8	3.8	3.6	3.7	AP > UP
task-1	0.11	0.20	0.13	0.27	-	2.0	2.0	2.3	2.4	AP < UP
task-2	0.21	0.11	0.20	0.17	-	3.0	3.0	2.9	2.9	-
task-3	0.10	0.03	0.10	0.07	-	3.0	3.0	2.7	2.9	AP > UP
task-4	0.94	0.97	0.80	0.90	AP > UP	5.0	5.0	4.3	4.8	AP > UP
task-5	0.87	0.84	0.90	0.82	-	5.0	5.0	4.6	4.8	AP > UP
task-6	0.81	0.81	0.85	0.75	-	5.0	5.0	4.7	4.6	AP > UP

Table 4: Task-specific evaluation results for user-involvement in planning on task performance. ACC_s denotes the strict accuracy of an action sequence, and ACC_e denotes the correctness of execution results. Bold fonts are used to highlight the best performance across conditions.

Tasks	ACC_s					ACC_e				
	AP-AE	AP-UE	UP-AE	UP-UE	Post-hoc results	AP-AE	AP-UE	UP-AE	UP-UE	Post-hoc results
Avg	**0.53**	0.46	0.46	0.48	-	0.54	0.53	0.47	**0.56**	-
task-1	0.00	0.00	0.10	**0.12**	AP < UP	0.00	0.00	0.10	**0.13**	AP < UP
task-2	**0.78**	0.64	0.61	0.57	-	**0.78**	0.72	0.66	0.75	-
task-3	**0.44**	0.12	0.36	0.28	-	0.44	0.42	0.36	**0.52**	-
task-4	**0.95**	0.89	0.75	0.82	AP > UP	**0.95**	0.89	0.75	0.82	AP > UP
task-5	**0.98**	0.91	0.90	0.90	-	**0.98**	0.91	0.92	0.90	-
task-6	0.05	**0.22**	0.02	0.18	-	0.06	**0.23**	0.03	0.22	-

calibrated trust in planning. To avoid a potential impact of user involvement in the execution stage, we conducted a two-way ANOVA test to confirm the findings. We only find a significant difference in task-4. Post-hoc Tukey HSD results show that participants in conditions with automatic planning (AP) showed significantly higher calibrated trust in planning outcomes than those in conditions with user-involved planning (UP). Thus, our experimental results do not support **H1**.

We noticed that the calibrated trust in planning is quite low in the high-risk tasks where all initial plans are imperfect. This indicates that many users across all conditions consider the generated plan trustworthy. On tasks with low risk, where the initial plan is of high quality, users achieved much higher calibrated trust in the planning outcome. We also find that conditions with user-involved execution (UE) show slightly higher CT_p in task-1 and task-4 than conditions with automatic execution (AE). With the same statistical test as **H1** analysis, such differences are not significant.

5.2.2 The Impact of User Involvement in Planning on Task Performance. To verify **H2**, we considered plan quality, the accuracy of action sequences (ACC_s), and the execution accuracy of the plan (ACC_e) for analysis. For plan quality (cf. Table 3), we conducted one-way ANOVA on plan quality considering the user involvement in the planning stage. We found that overall user involvement in the planning stage caused a decrease plan quality, especially on tasks with a perfect plan (*i.e.,* task 4, 5, 6, where plan quality = 5) and task-3. However, in task-1, where the original plan contains a grammar error, we find that user involvement in planning can improve the plan quality. As the action sequence accuracy (ACC_s) and execution accuracy (ACC_e) are not normally distributed, we conducted the Kruskal-Wallis H-test by considering the user involvement in the planning as the independent variable. The results are shown in Table 4. With further post-hoc Mann-Whitney tests, we found that while participants achieved a relatively higher accuracy of action sequences in condition AP-AE, the condition UP-UE achieved the best execution accuracy. In most tasks, condition UP-UE achieved better or compatible performance as other conditions. The only exception is task-4, where user involvement in the planning caused a significantly worse performance (both ACC_s and ACC_e). As user involvement does not consistently lead to improved performance, these results are not enough to support **H2**.

We found that in task-1 and task-6 most participants in the AP-AE condition achieved a very low success rate. This is mainly due to the imperfect plans and imperfect execution generated by LLMs. In task-1, the plan generated by LLMs includes one step which contains two actions to execute. Due to the inability to edit the plan, the LLM agent execution missed one transaction in conditions with automatic planning. In task-6, the plan generated by LLMs is correct. However, in the automatic execution of step 2 of the plan (*i.e.,* selecting an itinerary suggested), the LLM agent has a high probability of choosing an itinerary that does not match the task description. If the participants do not carefully check the task description, and correct this agent behavior, the execution results would be wrong. This also helps explain why user involvement substantially improves the task outcome accuracy in task-6. More details about tasks can be found in the appendix.

5.2.3 The Impact of User Involvement in Execution on Calibrated Trust in Execution Outcome. As we observe in Table 3, user involvement in planning can have some negative impact on the plan

Table 5: Task-specific evaluation results for user-involvement in execution on task performance. Bold fonts are used to highlight the best performance across conditions.

Tasks	ACC_s					ACC_e				
	AP-AE	AP-UE	UP-AE	UP-UE	Post-hoc results	AP-AE	AP-UE	UP-AE	UP-UE	Post-hoc results
Avg	**0.53**	0.46	0.50	0.51	-	0.54	0.53	0.50	**0.58**	-
task-1	0.00	0.00	0.10	**0.12**	-	0.00	0.00	0.10	**0.14**	-
task-2	**0.78**	0.64	0.67	0.62	-	**0.78**	0.72	0.69	**0.78**	-
task-3	**0.44**	0.12	0.42	0.29	AE > UE	0.44	0.42	0.42	**0.53**	-
task-4	**0.95**	0.89	0.94	0.88	-	**0.95**	0.89	0.94	0.88	-
task-5	0.98	0.91	**1.00**	0.98	-	0.98	0.91	**1.00**	0.98	-
task-6	0.05	**0.22**	0.02	0.19	AE < UE	0.06	**0.23**	0.04	**0.23**	AE < UE

quality, which further impacts the execution stage. To control such impact, we filtered out the tasks where plan quality decreased after user-involved planning in the analysis of user involvement in the execution stage. To verify **H3**, we conducted one-way ANOVA on calibrated trust in execution outcome (CT_e). The results are shown in Table 6. We found that user involvement in execution causes no significant difference across conditions. Thus, **H3** is not supported by our experimental results.

Table 6: Task-specific evaluation results for user-involvement in execution on calibrated trust in execution (CT_e). We also report the mean value for each measure on each condition.

Tasks	CT_e				
	AP-AE	AP-UE	UP-AE	UP-UE	Post-hoc results
Avg	**0.66**	0.65	0.64	0.65	-
task-1	0.48	0.44	**0.51**	0.49	-
task-2	0.78	**0.83**	0.71	0.80	-
task-3	0.51	0.41	**0.60**	0.47	-
task-4	**0.94**	0.92	0.88	0.86	-
task-5	0.89	0.92	**0.96**	0.94	-
task-6	0.37	0.38	0.28	**0.42**	-

5.2.4 The Impact of User Involvement on Overall Task Performance. Similar to the verification of **H3**, we excluded the tasks where plan quality decreased after user-involved planning in this analysis. As the plan is generated before user involvement in the execution, we only considered ACC_s and ACC_e in the analysis of user involvement in the execution stage. To verify **H4**, we conducted Kruskal-Wallis H-test by considering the user involvement in the execution as the independent variable. The results are shown in Table 5. With post-hoc Mann-Whitney tests, we found that user involvement in the execution stage showed significantly higher ACC_s and ACC_e in task-6 (where the LLM assistant mainly failed to choose the most suitable itinerary plan). We found that participants in the AP-AE condition achieved the best accuracy of action sequences (*i.e.,* ACC_s), and participants in condition UP-UE achieved the best execution accuracy (*i.e.,* ACC_e). In other words, the executed action sequence in condition AP-AE is more aligned with the ground truth action sequence annotated by the authors. However, with user involvement in the execution stage, participants in condition UP-UE have a better opportunity to obtain correct task outcomes by correcting potentially flawed actions. Such a difference is due to our measure of ACC_e, which tolerates the non-risky actions (*e.g.,* search flight) and failure of action predictions. In contrast, our measure of ACC_s considers this as a wrong action sequence. Thus, in task-3, even if we find automatic execution achieved significantly better ACC_s than user-involved execution, participants in condition AP-UE and UP-UE obtained comparable or higher execution accuracy (*i.e.,* ACC_e) than conditions with automatic execution. While user involvement shows some positive impact on the execution accuracy, such impact is not significant and consistent across all tasks. Only in task-6, where users can correct the errors made by the LLM agent (*i.e.,* the wrong itinerary selection mentioned in Section 5.2.2), user involvement in the execution shows a significant contribution to the task performance. Thus, these results are not enough to strictly support **H4**.

5.3 Exploratory Analysis

5.3.1 The Impact of Covariates. For further insights into all user factors on user trust and team performance, we calculated Spearman rank-order correlation coefficients for user trust, calibrated trust, risk perception, and task performance. As can be seen in Table 7, we found these covariates mainly show correlations with subjective user trust, calibrated trust in execution, and risk perception. First, all covariates (*i.e.,* user factors) positively correlated with user trust (four subscales in the trust in automation questionnaire [46]) and negatively correlated with perceived risk (average over six tasks). It indicates that users with more expertise or familiarity with such systems tend to trust the daily assistant and show less perceived risk when using it. Meanwhile, users with a general propensity to trust also tend to trust the AI system. Besides user trust, *Assistant Expertise* and *Propensity to Trust* show a significant negative correlation with calibrated trust in the execution outcome. Apart from the above correlation, these user factors do not significantly correlate with task performance measures or calibrated trust in the planning outcome.

5.3.2 Impact of Plan Quality and Risk Percetion. Besides the measures calculated over task batch, a task-level analysis of plan quality and risk perception can deepen our understanding of their impacts. Besides measures adopted in Table 7, we include task-level confidence in this analysis and exclude the subscales from the trust in automation questionnaire. Thus, we calculated Spearman rank-order correlation coefficients for task-level measures across all groups of participants (shown in Table 8). As we can see, both plan quality and risk perception significantly correlate with user trust, calibrated trust, task performance, and user confidence. The *plan*

Table 7: Spearman rank-order correlation coefficient for covariates level on dependent variables. All measures are calculated based on average over task batch. "†" and "††" indicate the effect of the variable is significant at the level of 0.05 and 0.0125, respectively.

Covariates		llm expertise		assistant expertise		Familiarity		Propensity to Trust	
Category	**Variables**	r	p	r	p	r	p	r	p
User Trust	Reliability/Competence	0.334	**.000**††	0.245	**.000**††	0.321	**.000**††	0.679	**.000**††
	Understanding/Predictability	0.307	**.000**††	0.164	**.010**††	0.208	**.001**††	0.380	**.000**††
	Intention of Developers	0.406	**.000**††	0.324	**.000**††	0.362	**.000**††	0.517	**.000**††
	Trust in Automation	0.380	**.000**††	0.278	**.000**††	0.356	**.000**††	0.698	**.000**††
Calibrated Trust	CT_p	0.053	.404	0.053	.402	0.056	.378	0.037	.566
	CT_e	-0.120	.059	-0.195	**.002**††	-0.032	.621	-0.174	**.006**††
Risk Perception	Perceived Risk	-0.187	**.003**††	-0.180	**.004**††	-0.237	**.000**††	-0.363	**.000**††
Task Performance	ACC_s	0.037	.560	-0.014	.823	0.110	.085	0.018	.772
	ACC_e	-0.000	.995	-0.037	.567	0.085	.184	0.007	.911
	Plan Quality	-0.035	.587	-0.037	.560	0.080	.211	-0.032	.611

Table 8: Task-specific spearman rank-order correlation coefficient for plan quality and risk perception. "†" and "††" indicate the effect of the variable is significant at the level of 0.05 and 0.0125, respectively.

Category	**Variables**	Plan Quality		Risk Perception	
		r	p	r	p
User Trust	Trust-p	0.056	.032†	-0.293	**.000**††
	Trust-e	0.258	**.000**††	-0.160	**.000**††
Calibrated Trust	CT_p	0.723	**.000**††	-0.102	**.000**††
	CT_e	0.221	**.000**††	0.000	.995
Task Performance	Plan Quality	-	-	-0.141	**.000**††
	ACC_e	0.400	**.000**††	-0.110	**.000**††
	ACC_s	0.446	**.000**††	-0.096	**.000**††
Confidence	Confidence-p	0.137	**.000**††	-0.532	**.000**††
	Confidence-e	0.225	**.000**††	-0.271	**.000**††

quality shows a significant positive correlation with most measures, which indicates users perform better and calibrate their trust in the LLM agents in tasks with a high-quality plan. By contrast, the *risk perceptions* shows a negative correlation with most measures and also a negative correlation with the plan quality.

5.3.3 Failure Analysis. As we find that plan quality substantially affects task execution accuracy, we look into task performance across different plan qualities. For the tasks with low-quality plans (plans fail to cover task information or plan with grammar errors, *i.e.,* plan quality=1, 2), the execution accuracy is 1.8%. While for tasks with a plan that may mislead action prediction (plan quality = 3, 4), our LLM agent-based daily assistant achieved 59% execution accuracy. The average execution accuracy for tasks with a high-quality plan (plan quality =5) is 66.7%.

We further check 717 tasks where a high-quality plan (plan quality = 5) is provided. Among them, 235 tasks provide wrong execution results. The main causes are: (1) Wrong action parameter prediction (48.9%). While action names match, one or more parameters mismatch the expected value at some step of the action sequence. (2) Invalid actions (48.5%). Given a perfect plan, the LLM agent failed to predict one valid action (failed to predict one action name or failed to predict some action parameter value) to execute in some steps. (3) Wrong action name prediction (2.6%). The generated action sequence has at least one action name prediction that mismatches the ground truth.

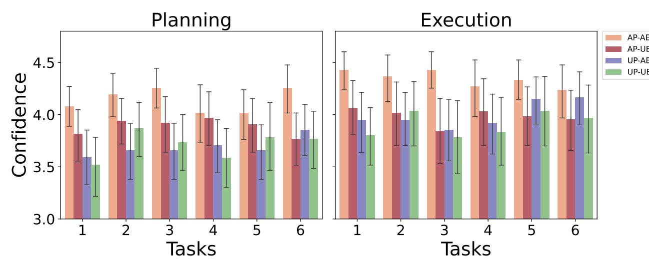

Figure 5: Bar plot for confidence dynamics, the x-axis denotes the task ordering index (shuffled for every participant). The error bars represent the 95% confidence interval.

5.3.4 Confidence Dynamics. To visualize the user confidence in the planning and execution stage, we draw point plots (see Figure 5) for user confidence in the task order. Overall, condition AP-AE shows the highest confidence in both the planning and execution stages. To verify the impact of user involvement in confidence, we adopted two-way ANOVA and post-hoc Tukey HSD test. We find that: (1) with user involvement in the planning, participants showed significantly lower confidence in planning (AP-AE > UP-AE, UP-UE); (2) with user involvement in the execution, participants showed a significantly lower confidence in execution (AP-AE > AP-UE, UP-UE). Meanwhile, users typically showed a higher confidence in the execution stage. Compared with conditions with automation execution (*i.e.,* condition AP-AE and UP-AE), the confidence gap narrows down in the conditions with user-involved execution (*i.e.,* condition AP-UE and UP-UE).

5.4 Analysis of Open Feedback

At the end of our study, we collected open feedback regarding the planning stage, execution stage, and any other feedback using the following question: '*Please share any comments, remarks or suggestions regarding the planning/execution stage of LLM Assistant*'

Table 9: Excerpts from participants' responses to open questions regarding opinion.

Opinion towards Planning	**Sentiment**	**Reason**
I really like how organized it is. The step by step and numerical planning allows it to make sense in a clear and structured way, meaning there is less room for errors or misinformation	Positive	helpful with reducing error
It was remarkable how quickly. It was able to achieve the goals which was set out in the tasks. I quite liked it I would definitely want something like this in my life as It would my my life much easier	Positive	Effective and make life easy
As I said previously, it's far, far too detailed in an unnecessary way. I'm not sure people need the entire plan of what the AI will do, as long as the job gets done.	Negative	too detailed
I found it really helpful, but made me slightly nervous thinking all my plans being successful are in the hands of ai tech	Mixed	helpful assistant, agency concerns
Opinion towards Execution	**Sentiment**	**Reason**
The execution stage was amazing. I feel like this could be the future and we wont need to call or talk to people to get this kind of thing done ever again.	Positive	promising future
The execution stage went smoothly, except for a few rare instances of an error response before also saying the AI's automatic reply (which was correct).	Mixed	Smooth user experience, error response
I found it clunky and nit that user friendly	Negative	clunky, not user-friendly
This bit is user friendly, but very robotic, which makes it difficult to trust	Mixed	user-friendly, distrust due to robotic nature

and '*Do you have any other comments, remarks or suggestions regarding the study?*'. Overall, we analyzed all the feedback based on user opinions (positive, negative, mixed, neutral) and their suggestions. In our analysis, we ignored all phrases without any useful information like 'None', 'N/A', and 'No comment'.

Feedback and Suggestions. While most comments tended to show positive opinions (more than 80%) towards LLM agents as daily assistants, there are also negative opinions regarding the difficulty, expertise, trust, etc. We provide example excerpts from participants in Table 9. Besides opinions towards the system, some participants also appreciated our user-centric setup:"*The study does a good job of emphasizing user experience by asking about perceptions of risk, trust, and confidence. This approach ensures that the evaluation is user-centric, which is important for assessing the real-world applicability of the LLM Assistant.*"

Some participants also provided suggestions on how to further improve the design of LLM agent-based daily assistants. Regarding the plan edit, participants hope we can provide more convenient edit operations like 'drop/drag' to adjust plan text ordering and 'undo' operation to tolerate unexpected mistakes. Some participants also found the plans too detailed, which could increase the cognitive load (cf. Table 9 except 3). As for the execution, many participants found it to be smooth. At the same time, they think additional verification in each step may further enhance the reliability of daily assistants: "*For the execution stage, I commend it for creating an input formatting box to execute the user's request validating each requirement.*" There are also comments about the whole plan-then-execute workflow: "*The planning was really challenging, and I mostly left the default plans (they looked fine). This worked in the main, but a couple clearly needed revisiting. I would approach this iteratively: plan, test, observe, back to planning, then another test, before reaching the desired outcome.*" Our findings suggest open research opportunities to explore more effective ways to provide an overview of plans that trade-off user cognitive load resulting from granular descriptions, with the need to provide details to help users identify flaws. For example, we can consider developing methods to interactively allow users to flesh out further details in a plan.

6 Discussion

6.1 Key Findings

Our experimental results show that user involvement in the plan-then-execute workflow with LLM agents can help fix imperfect plans in planning and wrong action predictions in the step-by-step execution. However, user involvement does not ensure a consistently positive impact on calibrated trust and overall task performance across different tasks.

User Involvement Fails to Calibrate User Trust. Overall, user involvement in the planning and execution does not significantly impact user trust and calibrated trust in planning and execution outcomes. As Table 3 shows, user involvement in planning can harm plan quality in tasks with a high-quality initial plan, which may potentially cause worse task performance in the subsequent execution stage. Our experimental results do not support **H1** or **H3**, which indicates user involvement does not necessarily help calibrate user trust in our study. Instead, with a task-specific correlation analysis (cf. Table 8), we found that the plan quality has a significant positive correlation with calibrated trust in both planning and execution outcomes. Combined with task-specific user trust and task-specific confidence, we can infer that users tend to trust the LLM agent overall. Such trust can be expected and calibrated in tasks with a high-quality plan. In contrast, users fail to calibrate their trust in the tasks where a low-quality plan is provided. A potential cause of such miscalibrated trust is the plausibility of plans generated by LLMs (i.e., plans that appear to be likely correct). In our study, all initial plans are formulated with a clear, logical structure, which covers most of the task requirements. At first glance, such high-quality text pieces seem quite plausible and trustworthy. We also received some open text feedback such as, — "The plans look nice, I do not find any space for improvement" and "the planning stage of the LLM assistant was helpful and trustworthy." Findings from recent work on LLM-assisted fact checking corroborate this, wherein authors found that convincing explanations provided by LLMs can cause over-reliance when LLMs are wrong [85].

User Involvement can Benefit Task Performance. User involvement in planning and execution can positively impact overall task performance, especially execution accuracy. As the results in Table 3 and Table 4 show, user involvement in planning can help address imperfect plans (*e.g.,* task-1 with grammar error). Doing so further contributes to improvements in the execution accuracy. After controlling the plan quality, we found that user involvement in the execution can provide the best execution accuracy among most tasks considered in our study (cf. Table 5). Based on the failure analysis (Section 5.3.3), LLM agents can make mistakes in executing high-quality plans, which can be attributed to prediction errors (*i.e.,* wrong action name or action parameters) and prediction failures (*i.e.,* failure to provide valid action prediction). In practice with deployed LLM services, there is no reliability guarantee for the generated plan in planning or predicted actions in execution. User involvement can play an important role in the plan quality control and risky action control, ensuring that only correct and safe actions are executed to obtain desirable task outcomes.

Other Findings. We also found some user factors and perceptions that affect user trust and task performance. As seen in Table 7, nearly all covariates show a significant positive correlation with user trust in the AI system. Some of these covariates also impact user trust in the planning and execution outcomes. Overall, these findings indicate that users who are familiar with such systems tend to show higher user trust. However, some factors also correlate negatively with the calibrated trust in the execution outcomes and risk perception of using the LLM agents as daily assistants. This reflects that these factors can cause miscalibrated trust and reduced risk perception when working with the LLM agent. While we found that risk perception negatively correlated with user trust, calibrated trust, task performance, and confidence (cf. Table 8), it does not mean risk perception is harmful in the human-LLM agent collaboration. The main cause is that users may only notice the risks of using LLM agents when the task is provided with a relatively low-quality plan. Risk perception is important to calibrate user trust in the planning and execution outcomes. Collaborative workflows should support users with the provision to take over control of planning and/or execution stages based on their perceived risk.

6.2 Implications

The Impact of Convincingly Wrong LLM Outcomes. As our study follows a plan-then-execute workflow for users to collaborate with LLM agents, users were not offered a chance to revise the plan after starting with execution. Users following a wrong plan can lead to negative outcomes. Combined with existing work on algorithm aversion [11] and the impact of negative first impressions on user trust [88], we can infer that such convincingly wrong content [85] can bias user trust and reliance towards the extremes. Before users take notice, they may develop an uncalibrated trust in the AI system, as observed through our findings in high-risk tasks (*i.e.,* tasks 1,2,3) and corroborating work by Si et al. [85]. As a result, users over-rely on AI assistance, which is misuse akin to behavior that resonates with algorithm appreciation [58]. Once users notice such phenomena, their trust in the LLM-based systems may sharply decrease, resulting in disuse due to algorithm aversion. This can be a result of the misalignment between perceived AI performance and actual AI performance. Existing human-AI collaboration literature has provided potential solutions for such problems, ranging from performance feedback interventions [29] to agreement-in-confidence heuristic [60, 74]. Future work can combine these insights to explore effective interventions for user trust calibration with convincingly wrong LLM outcomes.

Insights for Effective Collaboration with Plan-then-execute LLM Agents. Our work has important theoretical implications for effective human-AI collaboration with plan-then-execute LLM agents. On the one hand, user involvement can be necessary to achieve complementary team performance. Although LLM agents have shown promising planning and execution capabilities, they are never perfect due to probabilistic uncertainty. With user involvement in the planning, users can fix imperfect plans with grammar errors (cf. Table 3 task-1). With user involvement in the execution, users can fix uncertainty issues (*e.g.,* LLM agent predicts invalid actions) and prevent risky actions (*e.g.,* LLM agents choose an itinerary conflicting with task requirements, cf. Table 5, task-6). On the other hand, user involvement may also bring uncertainty and even harm LLM agent performance. In tasks where the LLM agent provides a high-quality plan (cf. task 4, 5, 6 in Table 3 and Table 4), user involvement can harm the plan quality, which further negatively impacts the execution accuracy. Moreover, user involvement in planning and execution poses a significantly higher cognitive load on users (cf. Figure 4) and negatively impacts user confidence (cf. Figure 5). Thus, too much human involvement in collaboration with plan-then-execute LLM agents can be undesirable. User involvement in the execution process brings more consistent benefits than user involvement in the planning stage. As suggested by the participants, iterative LLM agent simulation may be one potential way to decide when users should be involved. The LLM agent may first conduct a plan-then-execute round to obtain a clear plan and execution results. With humans checking the whole process and simulated outcomes, humans can decide whether to be involved in revising the plan or the execution process. In this way, we can minimize user involvement while keeping highly effective task outcomes through LLM agents.

Human Oversight and Designing Flexible Collaborative Workflows. In our study, we found that human oversight does not consistently lead to improved outcomes. One potential cause can be the disparity between the planning and execution of LLM agents. Specifically, it is unclear how one plan step will be transformed into one action. When users realize one plan step can be wrong during the execution stage, they may need to articulate it or manually override the agent action, posing a high cognitive load. Even worse, when users realize the LLM agent missed one action due to limited steps designed in the plan (in task-1), they do not have a chance to change the plan or add one extra step. To address such concerns, we may need a more flexible collaborative workflow where humans can fix planning and execution simultaneously. In this way, users can exercise more flexible control over the workflow and the task outcomes. For instance, the action prediction from the LLM agent can be provided along with each step in the planning stage. Users can thereby be informed of the potential impact of their edited plan, which provides more straightforward feedback and helps users adjust the plan according to the expected actions.

6.3 Caveats and Limitations

Limitations and Potentail Biases. To ensure reliable task outcomes, humans are expected to fix imperfect plans (*e.g.,* grammar errors, misleading action intents) in the planning stage. However, not everyone in conditions UP-AE and UP-UE noticed such grammar errors and split the plan in task-1. Similarly, not everyone in conditions AP-UE and UP-UE noticed that the LLM agent chose an itinerary that conflicted with task requirements. As discussed earlier, LLM agents can generate plausible plans, which may mislead user trust in the planning and execution outcomes. In that case, participants in our study may have easily ignored some convincingly wrong plan steps or execution actions. In our study, one primary step in the plan is only transformed into a single action. In practice, LLM agents can generate multiple actions for one specific goal. However, such action generation and execution modes are challenging for humans to get involved in and control, as the execution of the action sequences is automated by the LLM agent within one goal. Furthermore, using multiple actions to achieve one primary step (*i.e.,* goal) also results in higher task complexity and reduced task clarity, which may impact the task outcomes [23].

Transferability Concerns. Although we selected representative tasks for daily scenarios, our study may not be enough to cover all potential cases of daily assistance with LLM agents. Some task characteristics (*e.g.,* task complexity, time consumption) may also impact how users are willing to rely on AI assistance. Meanwhile, full control over the plan-then-execute LLM agents may not be desirable for some simple tasks (*e.g.,* setting alarms). Once the efforts to control/interact with LLM agents are greater than the efforts to execute the tasks themselves, users will be unwilling to adopt such "assistance." Future work can look into what daily user needs are suitable for LLM agents to support. In our study, the execution of plans is conducted in a simulation environment. While it has been proven to be effective in prior work of agent-based modeling and HCI studies [71], more work is needed to understand how execution of tasks in real-world environments with additional dependencies and complexities can influence our findings.

Participants in our study only followed a relatively fixed mode in collaboration with LLM agents, they can determine when to get involved in the planning and execution stages. The experimental conditions considered in our study range from full automation (*i.e.,* AP-AE) to full user control (*i.e.,* UP-UE). Such a setup provides good flexibility, which simulates real-world practice. Our findings and implications provide valuable insights to guide future research on human-AI collaboration with LLM agents.

7 Conclusion

This work empirically studied human-AI collaboration based on plan-then-execute LLM agents. Adopting such LLM agents in various everyday scenarios, we analyzed the impact of user involvement in the planning and execution stages on user trust and overall task performance. We provide various interactions in each stage to help users fix imperfect plans and modify execution outcomes. Our results suggest that the LLM agents can provide plausible text plans to cover task requirements, which can be convincingly wrong. As a result, users develop uncalibrated trust in the planning and execution outcomes, and user involvement in the planning and execution stages fails to calibrate user trust (**RQ1**). We also found that the plan quality substantially affects the subsequent execution accuracy. Thus, when user involvement in planning can fix imperfect plans, the overall task performance (*i.e.,* plan quality, accuracy of action sequence, and execution accuracy) gets improved. However, user involvement in planning can also harm task plan quality where the original plan is good to begin with. As a result, the LLM agents demonstrate worse task performance in these tasks. In contrast, user involvement in execution brings a more stable positive impact on task performance (**RQ2**). Our results suggest that plausible but wrong LLM outcomes can be detrimental to user trust calibration and overall task performance. We discussed the impact of convincingly wrong LLM outcomes and provided potential solutions and insights for future work. Furthermore, we synthesized key insights for better control and effective collaboration with plan-then-execute LLM agents. We also shed light on opportunities to design flexible collaborative workflows with human oversight for effective collaboration with LLM agents.

Our results indicate that user involvement in the LLM agent workflow can be important in ensuring reliable task outcomes. Future work can further investigate how to detect and handle plausible but imperfect LLM outcomes and design effective interventions to fix such problems. We hope that our key findings and implications reported in this work will inspire further research on human-AI collaboration with LLM agents.

Acknowledgments

This work was partially supported by the Delft Design@Scale AI Lab, the 4TU.CEE UNCAGE project, the Convergence Flagship "ProtectMe" project, by the Australian Research Council (ARC) Training Centre for Information Resilience (Grant No. IC200100022), and by an ARC Future Fellowship Project (Grant No. FT240100022). We made use of the Dutch national e-infrastructure with the support of the SURF Cooperative using grant no. EINF-5571 and EINF-9738. We finally thank all participants from Prolific and experts from our department.

References

[1] Stefano V Albrecht and Peter Stone. 2018. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence* 258 (2018), 66–95.

[2] Nikola Banovic, Zhuoran Yang, Aditya Ramesh, and Alice Liu. 2023. Being trustworthy is not enough: How untrustworthy artificial intelligence (AI) can deceive the end-users and gain their trust. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW1 (2023), 1–17.

[3] Gagan Bansal, Besmira Nushi, Ece Kamar, Walter S Lasecki, Daniel S Weld, and Eric Horvitz. 2019. Beyond accuracy: The role of mental models in human-AI team performance. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, Vol. 7. 2–11.

[4] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258* (2021).

[5] Zana Buçinca, Maja Barbara Malaya, and Krzysztof Z Gajos. 2021. To trust or to think: cognitive forcing functions can reduce overreliance on AI in AI-assisted decision-making. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (2021), 1–21.

[6] Chun-Wei Chiang and Ming Yin. 2022. Exploring the Effects of Machine Learning Literacy Interventions on Laypeople's Reliance on Machine Learning Models. In *IUI 2022: 27th International Conference on Intelligent User Interfaces, Helsinki,*

Finland, March 22 - 25, 2022, Giulio Jacucci, Samuel Kaski, Cristina Conati, Simone Stumpf, Tuukka Ruotsalo, and Krzysztof Gajos (Eds.). ACM, 148–161.
[7] Chun-Wei Chiang and Ming Yin. 2021. You'd better stop! Understanding human reliance on machine learning models under covariate shift. In *Proceedings of the 13th ACM Web Science Conference 2021*. 120–129.
[8] Michael Chromik, Malin Eiband, Felicitas Buchner, Adrian Krüger, and Andreas Butz. 2021. I think i get your point, AI! the illusion of explanatory depth in explainable AI. In *Proceedings of the 26th International Conference on Intelligent User Interfaces*. 307–317.
[9] Lacey Colligan, Henry WW Potts, Chelsea T Finn, and Robert A Sinkin. 2015. Cognitive workload changes for nurses transitioning from a legacy system with paper documentation to a commercial electronic health record. *International journal of medical informatics* 84, 7 (2015), 469–476.
[10] Jason P Davis, Kathleen M Eisenhardt, and Christopher B Bingham. 2007. Developing theory through simulation methods. *Academy of management review* 32, 2 (2007), 480–499.
[11] Berkeley J Dietvorst, Joseph P Simmons, and Cade Massey. 2015. Algorithm aversion: people erroneously avoid algorithms after seeing them err. *Journal of experimental psychology: General* 144, 1 (2015), 114.
[12] Berkeley J Dietvorst, Joseph P Simmons, and Cade Massey. 2018. Overcoming algorithm aversion: People will use imperfect algorithms if they can (even slightly) modify them. *Management science* 64, 3 (2018), 1155–1170.
[13] Shi Dong, Ping Wang, and Khushnood Abbas. 2021. A survey on deep learning and its applications. *Computer Science Review* 40 (2021), 100379.
[14] Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608* (2017).
[15] Mary T Dzindolet, Scott A Peterson, Regina A Pomranky, Linda G Pierce, and Hall P Beck. 2003. The role of trust in automation reliance. *International journal of human-computer studies* 58, 6 (2003), 697–718.
[16] Sven Eckhardt, Niklas Kühl, Mateusz Dolata, and Gerhard Schwabe. 2024. A Survey of AI Reliance. *arXiv preprint arXiv:2408.03948* (2024).
[17] Alexander Erlei, Abhinav Sharma, and Ujwal Gadiraju. 2024. Understanding Choice Independence and Error Types in Human-AI Collaboration. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–19.
[18] Shaoyang Fan, Ujwal Gadiraju, Alessandro Checco, and Gianluca Demartini. 2020. Crowdco-op: Sharing risks and rewards in crowdsourcing. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW2 (2020), 1–24.
[19] Franz Faul, Edgar Erdfelder, Axel Buchner, and Albert-Georg Lang. 2009. Statistical power analyses using G* Power 3.1: Tests for correlation and regression analyses. *Behavior research methods* 41, 4 (2009), 1149–1160.
[20] Tharindu Fernando, Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. 2021. Deep learning for medical anomaly detection–a survey. *ACM Computing Surveys (CSUR)* 54, 7 (2021), 1–37.
[21] Riccardo Fogliato, Alexandra Chouldechova, and Zachary Lipton. 2021. The impact of algorithmic risk assessments on human predictions and its analysis via crowdsourcing studies. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–24.
[22] Andreas Fügener, Jörn Grahl, Alok Gupta, and Wolfgang Ketter. 2022. Cognitive challenges in human–artificial intelligence collaboration: Investigating the path toward productive delegation. *Information Systems Research* 33, 2 (2022), 678–696.
[23] Ujwal Gadiraju, Jie Yang, and Alessandro Bozzon. 2017. Clarity is a worthwhile quality: On the role of task clarity in microtask crowdsourcing. In *Proceedings of the 28th ACM conference on hypertext and social media*. 5–14.
[24] Florian Geissler, Karsten Roscher, and Mario Trapp. 2024. Concept-Guided LLM Agents for Human-AI Safety Codesign. In *Proceedings of the AAAI Symposium Series*, Vol. 3. 100–104.
[25] Zoubin Ghahramani. 2015. Probabilistic machine learning and artificial intelligence. *Nature* 521, 7553 (2015), 452–459.
[26] Ben Green and Yiling Chen. 2021. Algorithmic risk assessments can alter human decision-making processes in high-stakes government contexts. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–33.
[27] Gaole He, Nilay Aishwarya, and Ujwal Gadiraju. 2025. Is Conversational XAI All You Need? Human-AI Decision Making With a Conversational XAI Assistant. In *Proceedings of the 30th International Conference on Intelligent User Interfaces*.
[28] Gaole He, Abri Bharos, and Ujwal Gadiraju. 2024. To Err Is AI! Debugging as an Intervention to Facilitate Appropriate Reliance on AI Systems. In *Proceedings of the 35th ACM Conference on Hypertext and Social Media*. 98–105.
[29] Gaole He, Lucie Kuiper, and Ujwal Gadiraju. 2023. Knowing About Knowing: An Illusion of Human Competence Can Hinder Appropriate Reliance on AI Systems. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–18.
[30] Zeyu He, Chieh-Yang Huang, Chien-Kuang Cornelia Ding, Shaurya Rohatgi, and Ting-Hao Kenneth Huang. 2024. If in a Crowdsourced Data Annotation Pipeline, a GPT-4. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–25.
[31] Patrick Hemmer, Sebastian Schellhammer, Michael Vössing, Johannes Jakubik, and Gerhard Satzger. 2022. Forming Effective Human-AI Teams: Building Machine Learning Models that Complement the Capabilities of Multiple Experts. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, Lud De Raedt (Ed.). International Joint Conferences on Artificial Intelligence Organization, 2478–2484. doi:10.24963/ijcai.2022/344 Main Track.
[32] Patrick Hemmer, Max Schemmer, Niklas Kühl, Michael Vössing, and Gerhard Satzger. 2024. Complementarity in Human-AI Collaboration: Concept, Sources, and Evidence. *arXiv preprint arXiv:2404.00029* (2024).
[33] Patrick Hemmer, Max Schemmer, Michael Vössing, and Niklas Kühl. 2021. Human-AI Complementarity in Hybrid Intelligence Systems: A Structured Literature Review. *PACIS* (2021), 78.
[34] Patrick Hemmer, Monika Westphal, Max Schemmer, Sebastian Vetter, Michael Vössing, and Gerhard Satzger. 2023. Human-AI collaboration: the effect of AI delegation on human task performance and task satisfaction. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*. 453–463.
[35] Yoyo Tsung-Yu Hou and Malte F Jung. 2021. Who is the expert? Reconciling algorithm aversion and algorithm appreciation in AI-supported decision making. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–25.
[36] Shijue Huang, Wanjun Zhong, Jianqiao Lu, Qi Zhu, Jiahui Gao, Weiwen Liu, Yutai Hou, Xingshan Zeng, Yasheng Wang, Lifeng Shang, Xin Jiang, Ruifeng Xu, and Qun Liu. 2024. Planning, Creation, Usage: Benchmarking LLMs for Comprehensive Tool Utilization in Real-World Complex Scenarios. arXiv:2401.17167 [cs.CL]
[37] Alon Jacovi, Ana Marasović, Tim Miller, and Yoav Goldberg. 2021. Formalizing trust in artificial intelligence: Prerequisites, causes and goals of human trust in AI. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*. 624–635.
[38] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *Comput. Surveys* 55, 12 (2023), 1–38.
[39] Robin Jia and Percy Liang. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2021–2031.
[40] Jialun Aaron Jiang, Kandrea Wade, Casey Fiesler, and Jed R Brubaker. 2021. Supporting serendipity: Opportunities and challenges for Human-AI Collaboration in qualitative analysis. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (2021), 1–23.
[41] Patricia K Kahr, Gerrit Rooks, Martijn C Willemsen, and Chris CP Snijders. 2024. Understanding Trust and Reliance Development in AI Advice: Assessing Model Accuracy, Model Explanations, and Experiences from Previous Interactions. *ACM Transactions on Interactive Intelligent Systems* (2024).
[42] Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Kaya Stechly, Mudit Verma, Siddhant Bhambri, Lucas Saldyt, and Anil Murthy. 2024. LLMs Can't Plan, But Can Help Planning in LLM-Modulo Frameworks. *arXiv preprint arXiv:2402.01817* (2024).
[43] Davinder Kaur, Suleyman Uslu, Kaley J Rittichier, and Arjan Durresi. 2022. Trustworthy artificial intelligence: a review. *ACM computing surveys (CSUR)* 55, 2 (2022), 1–38.
[44] Veton Kepuska and Gamal Bohouta. 2018. Next-generation of virtual personal assistants (microsoft cortana, apple siri, amazon alexa and google home). In *2018 IEEE 8th annual computing and communication workshop and conference (CCWC)*. IEEE, 99–103.
[45] Sunnie S. Y. Kim, Q. Vera Liao, Mihaela Vorvoreanu, Stephanie Ballard, and Jennifer Wortman Vaughan. 2024. "I'm Not Sure, But...": Examining the Impact of Large Language Models' Uncertainty Expression on User Reliance and Trust. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency* (Rio de Janeiro, Brazil) *(FAccT '24)*. Association for Computing Machinery, New York, NY, USA, 822–835. doi:10.1145/3630106.3658941
[46] Moritz Körber. 2019. Theoretical considerations and development of a questionnaire to measure trust in automation. In *Proceedings of the 20th Congress of the International Ergonomics Association (IEA 2018) Volume VI: Transport Ergonomics and Human Factors (TEHF), Aerospace Human Factors and Ergonomics 20*. Springer, 13–30.
[47] Vivian Lai, Samuel Carton, Rajat Bhatnagar, Q Vera Liao, Yunfeng Zhang, and Chenhao Tan. 2022. Human-ai collaboration via conditional delegation: A case study of content moderation. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–18.
[48] Vivian Lai, Chacha Chen, Alison Smith-Renner, Q. Vera Liao, and Chenhao Tan. 2023. Towards a Science of Human-AI Decision Making: An Overview of Design Space in Empirical Human-Subject Studies. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency* (Chicago, IL, USA) *(FAccT '23)*. Association for Computing Machinery, New York, NY, USA, 1369–1385. doi:10.1145/3593013.3594087
[49] Vivian Lai, Han Liu, and Chenhao Tan. 2020. " Why is' Chicago'deceptive?" Towards Building Model-Driven Tutorials for Humans. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.
[50] John Lee and Neville Moray. 1992. Trust, control strategies and allocation of function in human-machine systems. *Ergonomics* 35, 10 (1992), 1243–1270.

[51] John D Lee and Katrina A See. 2004. Trust in automation: Designing for appropriate reliance. *Human factors* 46, 1 (2004), 50–80.

[52] Zhuoyan Li and Ming Yin. [n. d.]. Utilizing Human Behavior Modeling to Manipulate Explanations in AI-Assisted Decision Making: The Good, the Bad, and the Scary. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[53] Q Vera Liao and S Shyam Sundar. 2022. Designing for responsible trust in AI systems: A communication perspective. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. 1257–1268.

[54] Q Vera Liao and Jennifer Wortman Vaughan. 2023. AI Transparency in the Age of LLMs: A Human-Centered Research Roadmap. *arXiv preprint arXiv:2306.01941* (2023).

[55] Gabriel Lima, Nina Grgić-Hlača, and Meeyoung Cha. 2021. Human perceptions on moral responsibility of AI: A case study in AI-assisted bail decision-making. In *Proceedings of the 2021 CHI conference on human factors in computing systems*. 1–17.

[56] Jessy Lin, Nicholas Tomlin, Jacob Andreas, and Jason Eisner. 2024. Decision-oriented dialogue for human-ai collaboration. *Transactions of the Association for Computational Linguistics* 12 (2024), 892–911.

[57] Han Liu, Vivian Lai, and Chenhao Tan. 2021. Understanding the effect of out-of-distribution examples and interactive explanations on human-ai decision making. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–45.

[58] Jennifer M Logg, Julia A Minson, and Don A Moore. 2019. Algorithm appreciation: People prefer algorithmic to human judgment. *Organizational Behavior and Human Decision Processes* 151 (2019), 90–103.

[59] Zhuoran Lu, Dakuo Wang, and Ming Yin. 2024. Does more advice help? the effects of second opinions in AI-assisted decision making. *Proceedings of the ACM on Human-Computer Interaction* 8, CSCW1 (2024), 1–31.

[60] Zhuoran Lu and Ming Yin. 2021. Human Reliance on Machine Learning Models When Performance Feedback is Limited: Heuristics and Risks. In *CHI '21: CHI Conference on Human Factors in Computing Systems, Virtual Event / Yokohama, Japan, May 8-13, 2021*, Yoshifumi Kitamura, Aaron Quigley, Katherine Isbister, Takeo Igarashi, Pernille Bjørn, and Steven Mark Drucker (Eds.). ACM, 78:1–78:16.

[61] Brian Lubars and Chenhao Tan. 2019. Ask not what AI can do, but what AI should do: Towards a framework of task delegability. *Advances in neural information processing systems* 32 (2019).

[62] Shuai Ma, Qiaoyi Chen, Xinru Wang, Chengbo Zheng, Zhenhui Peng, Ming Yin, and Xiaojuan Ma. 2024. Towards human-ai deliberation: Design and evaluation of llm-empowered deliberative ai for ai-assisted decision-making. *arXiv preprint arXiv:2403.16812* (2024).

[63] Shuai Ma, Ying Lei, Xinru Wang, Chengbo Zheng, Chuhan Shi, Ming Yin, and Xiaojuan Ma. 2023. Who should i trust: Ai or myself? leveraging human and ai correctness likelihood to promote appropriate trust in ai-assisted decision-making. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–19.

[64] Shuai Ma, Xinru Wang, Ying Lei, Chuhan Shi, Ming Yin, and Xiaojuan Ma. 2024. "Are You Really Sure?" Understanding the Effects of Human Self-Confidence Calibration in AI-Assisted Decision Making. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–20.

[65] David Madras, Toni Pitassi, and Richard Zemel. 2018. Predict responsibly: improving fairness and accuracy by learning to defer. *Advances in neural information processing systems* 31 (2018).

[66] Hasan Mahmud, AKM Najmul Islam, Syed Ishtiaque Ahmed, and Kari Smolander. 2022. What influences algorithmic decision-making? A systematic literature review on algorithm aversion. *Technological Forecasting and Social Change* 175 (2022), 121390.

[67] Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 3428–3448. doi:10.18653/v1/P19-1334

[68] Siddharth Mehrotra, Chadha Degachi, Oleksandra Vereschak, Catholijn M Jonker, and Myrthe L Tielman. 2024. A Systematic Review on Fostering Appropriate Trust in Human-AI Interaction: Trends, Opportunities and Challenges. *ACM Journal on Responsible Computing* 1, 4 (2024), 1–45.

[69] George A Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review* 63, 2 (1956), 81.

[70] Harikrishna Narasimhan, Wittawat Jitkrittum, Aditya K Menon, Ankit Rawat, and Sanjiv Kumar. 2022. Post-hoc estimators for learning to defer to an expert. *Advances in Neural Information Processing Systems* 35 (2022), 29292–29304.

[71] Judith S Olson and Wendy A Kellogg. 2014. *Ways of Knowing in HCI*. Vol. 2. Springer.

[72] Behrooz Omidvar Tehrani and Anmol Anubhai. 2024. Evaluating Human-AI Partnership for LLM-based Code Migration. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. 1–8.

[73] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*. 1–22.

[74] Niccolò Pescetelli and Nicholas Yeung. 2021. The role of decision confidence in advice-taking and trust formation. *Journal of Experimental Psychology: General* 150, 3 (2021), 507.

[75] Marc Pinski, Martin Adam, and Alexander Benlian. 2023. AI knowledge: Improving AI delegation through human enablement. In *Proceedings of the 2023 CHI conference on human factors in computing systems*. 1–17.

[76] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and Sundaraja S Iyengar. 2018. A survey on deep learning: Algorithms, techniques, and applications. *ACM computing surveys (CSUR)* 51, 5 (2018), 1–36.

[77] Charvi Rastogi, Yunfeng Zhang, Dennis Wei, Kush R Varshney, Amit Dhurandhar, and Richard Tomsett. 2022. Deciding fast and slow: The role of cognitive biases in ai-assisted decision-making. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW1 (2022), 1–22.

[78] Amy Rechkemmer and Ming Yin. 2022. When confidence meets accuracy: Exploring the effects of multiple performance indicators on trust in machine learning models. In *Proceedings of the 2022 chi conference on human factors in computing systems*. 1–14.

[79] Sara Salimzadeh, Gaole He, and Ujwal Gadiraju. 2023. A Missing Piece in the Puzzle: Considering the Role of Task Complexity in Human-AI Decision Making. In *Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization*. 215–227.

[80] Sara Salimzadeh, Gaole He, and Ujwal Gadiraju. 2024. Dealing with Uncertainty: Understanding the Impact of Prognostic Versus Diagnostic Tasks on Trust and Reliance in Human-AI Decision Making. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–17.

[81] Max Schemmer, Patrick Hemmer, Niklas Kühl, Carina Benz, and Gerhard Satzger. 2022. Should I Follow AI-based Advice? Measuring Appropriate Reliance in Human-AI Decision-Making. In *ACM Conference on Human Factors in Computing Systems (CHI'22), Workshop on Trust and Reliance in AI-Human Teams (trAIt)*.

[82] Max Schemmer, Niklas Kuehl, Carina Benz, Andrea Bartos, and Gerhard Satzger. 2023. Appropriate reliance on AI advice: Conceptualization and the effect of explanations. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*. 410–422.

[83] Ashish Sharma, Inna W Lin, Adam S Miner, David C Atkins, and Tim Althoff. 2023. Human–AI collaboration enables more empathic conversations in text-based peer-to-peer mental health support. *Nature Machine Intelligence* 5, 1 (2023), 46–57.

[84] Hua Shen, Chieh-Yang Huang, Tongshuang Wu, and Ting-Hao Kenneth Huang. 2023. ConvXAI: Delivering heterogeneous AI explanations via conversations to support human-AI scientific writing. In *Companion Publication of the 2023 Conference on Computer Supported Cooperative Work and Social Computing*. 384–387.

[85] Chenglei Si, Navita Goyal, Tongshuang Wu, Chen Zhao, Shi Feng, Hal Daumé Iii, and Jordan Boyd-Graber. 2024. Large Language Models Help Humans Verify Truthfulness–Except When They Are Convincingly Wrong. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 1459–1474.

[86] Dylan Slack, Satyapriya Krishna, Himabindu Lakkaraju, and Sameer Singh. 2023. Explaining machine learning models with interactive natural language conversations using TalkToModel. *Nature Machine Intelligence* 5, 8 (2023), 873–883.

[87] Suzanne Tolmeijer, Markus Christen, Serhiy Kandul, Markus Kneer, and Abraham Bernstein. 2022. Capable but amoral? Comparing AI and human expert collaboration in ethical decision making. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–17.

[88] Suzanne Tolmeijer, Ujwal Gadiraju, Ramya Ghantasala, Akshit Gupta, and Abraham Bernstein. 2021. Second chance for a first impression? Trust development in intelligent system interaction. In *Proceedings of the 29th ACM Conference on user modeling, adaptation and personalization*. 77–87.

[89] Richard Tomsett, Alun Preece, Dave Braines, Federico Cerutti, Supriyo Chakraborty, Mani Srivastava, Gavin Pearson, and Lance Kaplan. 2020. Rapid trust calibration through interpretable and uncertainty-aware AI. *Patterns* 1, 4 (2020), 100049.

[90] Amos Tversky and Daniel Kahneman. 1991. Loss aversion in riskless choice: A reference-dependent model. *The quarterly journal of economics* 106, 4 (1991), 1039–1061.

[91] Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. 2023. On the planning abilities of large language models-a critical investigation. *Advances in Neural Information Processing Systems* 36 (2023), 75993–76005.

[92] Helena Vasconcelos, Matthew Jörke, Madeleine Grunde-McLaughlin, Tobias Gerstenberg, Michael S Bernstein, and Ranjay Krishna. 2023. Explanations can

reduce overreliance on ai systems during decision-making. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW1 (2023), 1–38.

[93] Oleksandra Vereschak, Gilles Bailly, and Baptiste Caramiaux. 2021. How to evaluate trust in AI-assisted decision making? A survey of empirical methodologies. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–39.

[94] Michael Vössing, Niklas Kühl, Matteo Lind, and Gerhard Satzger. 2022. Designing transparency for effective human-AI collaboration. *Information Systems Frontiers* 24, 3 (2022), 877–895.

[95] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science* 18, 6 (2024), 186345.

[96] Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2609–2634.

[97] Xinru Wang, Hannah Kim, Sajjadur Rahman, Kushan Mitra, and Zhengjie Miao. 2024. Human-LLM collaborative annotation through effective verification of LLM labels. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–21.

[98] Xinru Wang and Ming Yin. 2021. Are Explanations Helpful? A Comparative Study of the Effects of Explanations in AI-Assisted Decision-Making. In *26th International Conference on Intelligent User Interfaces*. 318–328.

[99] Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Ma, Yitao Liang, and Team CraftJarvis. 2023. Describe, explain, plan and select: interactive planning with large language models enables open-world multi-task agents. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*. 34153–34189.

[100] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682* (2022).

[101] Robert E Wood. 1986. Task complexity: Definition of the construct. *Organizational behavior and human decision processes* 37, 1 (1986), 60–82.

[102] Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022. Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In *Proceedings of the 2022 CHI conference on human factors in computing systems*. 1–22.

[103] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2023. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864* (2023).

[104] Yanming Yang, Xin Xia, David Lo, and John Grundy. 2022. A survey on deep learning for software engineering. *ACM Computing Surveys (CSUR)* 54, 10s (2022), 1–73.

[105] Ming Yin, Jennifer Wortman Vaughan, and Hanna Wallach. 2019. Understanding the effect of accuracy on trust in machine learning models. In *Proceedings of the 2019 chi conference on human factors in computing systems*. 1–12.

[106] Yunfeng Zhang, Q. Vera Liao, and Rachel K. E. Bellamy. 2020. Effect of confidence and explanation on accuracy and trust calibration in AI-assisted decision making. In *FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020*, Mireille Hildebrandt, Carlos Castillo, L. Elisa Celis, Salvatore Ruggieri, Linnet Taylor, and Gabriela Zanfir-Fortuna (Eds.). ACM, 295–305.

[107] Zhiping Zhang, Michelle Jia, Hao-Ping Lee, Bingsheng Yao, Sauvik Das, Ada Lerner, Dakuo Wang, and Tianshi Li. 2024. "It's a Fair Game", or Is It? Examining How Users Navigate Disclosure Risks and Benefits When Using LLM-Based Conversational Agents. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–26.

[108] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223* (2023).

[109] Qingxiao Zheng, Zhongwei Xu, Abhinav Choudhary, Yuting Chen, Yongming Li, and Yun Huang. 2023. Synergizing human-AI agency: a guide of 23 heuristics for service co-creation with LLM-based agents. *arXiv preprint arXiv:2310.15065* (2023).

[110] Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. 2024. Toolqa: A dataset for llm question answering with external tools. *Advances in Neural Information Processing Systems* 36 (2024).

A Details about Tasks

Here, we provide more details about the selected task used in our study. For each task, we provide the automatic planning (*i.e.,* step-wise text plan) and execution (*i.e.,* action sequence) generated with our plan-then-execute LLM agent.

Task-1: My account ID is 54321, and the password is PWD2023. I plan to make two foreign exchange transactions. The first is to buy 10,000 euros (with USD), and the second is to sell 5,000 US dollars (to EUR). Please help me operate.

Automatic Plan:

- 1. Log in to user account
  - 1.1 Obtain account login information (Account ID: 54321, Account Password: PWD2023)
  - 1.2 Log in to the account
  - 1.3 Confirm successful login
- 2. Conduct foreign exchange transactions
  - 2.1 Buy euros
    - 2.1.1 Obtain information for buying euros (Currency Type: EUR, Purchase Amount: 10000)
    - 2.1.2 Buy the specified amount of euros
    - 2.1.3 Confirm successful euro purchase
  - 2.2 Sell US dollars
    - 2.2.1 Obtain information for selling US dollars (Currency Type: USD, Sell Amount: 5000)
    - 2.2.2 Check the US dollar holdings
    - 2.2.3 Obtain US dollar holdings information (Foreign Exchange Holdings Information: Obtained US dollar holdings information)
    - 2.2.4 Sell the specified amount of US dollars
    - 2.2.5 Confirm successful US dollar sale

Automatic execution:

- bank_account_login(account=54321, passwd=PWD2023)
- buy_currency(account=54321, passwd=PWD2023, sourceCurrency=USD, targetCurrency=EUR, amount=10000)

Summary: The plan is imperfect. Users are supposed to split the primary step 2 into two primary steps. Then, the LLM agent can conduct two transactions instead of one.

Task-2: Please inquire about the current debt amount of my credit card with the last five digits 12345, and deduct the corresponding 12000 USD from my savings card number 6212345678900011 to repay this debt, then help me check the amount of the outstanding bill for the same credit card within 30 days after today.

Automatic Plan:

- 1. Inquire about the credit card debt amount
  - 1.1 Obtain credit card information (Account ID: Account ID, Account Password: Account Password, Credit Card: Last five digits 12345, Verification Code: Verification Code)
  - 1.2 Log in to the banking service platform
  - 1.3 Inquire about the current credit card debt amount
  - 1.4 Confirm the debt amount inquiry is successful and obtain the amount (Debt Amount: Inquired Debt Amount)
- 2. Deduct funds to repay the credit card debt
  - 2.1 Obtain savings card information (Savings Card Number: 6212345678900011)
  - 2.2 Transfer from savings card to credit card
  - 2.3 Confirm the transfer is successful
- 3. Check the credit card's future outstanding bills
  - 3.1 Obtain date information (Date Range: Within 30 days after today)
  - 3.2 Inquire about the credit card's outstanding bill amount within the specified date range
  - 3.3 Confirm the future outstanding bill amount inquiry is successful and obtain the amount (Outstanding Bill Amount: Inquired Outstanding Bill Amount)

Automatic execution:

- check_balance(card=12345)
- pay_credit_card(source_card=6212345678900011, target_card =12345, amount=12000)
- check_credit_card_bills(card=12345, period=30)

Summary: The authors need to revise the plan and add one primary step to search for the credit card, which ends with '12345'. Due to missing this step, the LLM agent can not correctly execute all tasks because the correct credit card ID (*i.e.,* 4980981092312345) is missing.

Task-3: I need to schedule a repair for my TV at 6 PM tomorrow evening. The brand is Sony, model X800H, and there is an issue with the screen. Please book the repair service and tell me the reservation number.

Automatic Plan:

- 1. Obtain user contact information
  - 1.1 Obtain user's name (Name: User's name)
  - 1.2 Obtain user's contact number (Contact number: User's phone number)
  - 1.3 Obtain the installation address of the user's TV (Installation address: User's address)
- 2. Find repair service
  - 2.1 Determine the repair service provider
    - 2.1.1 Search for authorized Sony TV repair service providers
    - 2.1.2 Select the nearest service provider to the user (Service provider list: Authorized Sony TV repair service provider list, User address: User's address)
  - 2.2 Obtain the contact information of the repair service provider (Service provider contact information: Nearest service provider's contact information to the user)
- 3. Book repair service
  - 3.1 Contact the repair service provider
    - 3.1.1 Provide the brand, model, and issue description of the TV (Brand: Sony, Model: X800H, Issue description: Screen issue)
    - 3.1.2 Provide the user's contact information (Name: User's name, Contact number: User's phone number, Installation address: User's address)
    - 3.1.3 Obtain the appointment time (Appointment time: Tomorrow's date 18:00:00)
    - 3.1.4 Book the repair service

  - 3.2 Obtain the reservation number (Reservation status: Repair request status, Reservation number: Repair request number)
- 4. Notify the user
  - 4.1 Send the reservation number and the contact information of the repair service provider to the user (Reservation number: Repair request number, Service provider contact information: Repair service provider's contact information)

Automatic execution:

- obtain_user_info()
- search_service_provider(brand=Sony)
- appliance_repair_request(applianceModel=X800H, applianceType=TV, appointment_time=Tomorrow 6 PM,issueDescription =Screen issue, servicepointID=[Sony01, Sony03])
- appliance_repair_request(applianceModel=X800H, applianceType=TV, appointment_time=Tomorrow 6 PM,issueDescription =Screen issue, servicepointID=[Sony01, Sony03])

Summary: While the first three primary steps are clear, the last primary step is not aligned with any provided actions. Users are supposed to delete the last primary step or override the potentially unnecessary actions after the first three steps.

Task-4: I need to set an alarm for every weekday morning at 7:30, and then cancel the alarm for Thursday, changing it to 8:00 in the evening.

Automatic Plan:

- 1. Set the alarm for every day
  - 1.1 Get the alarm setting information (Time: 07:30 AM, Frequency: Monday to Friday)
  - 1.2 Set the alarm
  - 1.3 Confirm whether the alarm is set successfully
- 2. Cancel the Thursday alarm
  - 2.1 Get the information of the alarm to be cancelled (Time: 07:30 AM, Frequency: Thursday)
  - 2.2 Cancel the alarm
  - 2.3 Confirm whether the alarm is cancelled successfully
- 3. Set the new alarm time
  - 3.1 Get the information of the alarm to be adjusted (New Time: 08:00 PM, Frequency: Thursday)
  - 3.2 Set the new alarm
  - 3.3 Confirm whether the new alarm is set successfully

Automatic execution:

- create_alarm(hour=7, minute=30, repeat=Weekly, frequency =Weekday)
- cancel_alarm(hour=7, minute=30, repeat=Weekly, frequency =Thursday)
- create_alarm(hour=20, minute=0, repeat=Weekly, frequency =Thursday)

Summary: Both the automatic planning and execution are correct.

Task-5: I have an important meeting to attend next Wednesday, and I need to book a flight ticket from London to Amsterdam for tomorrow, it must be a morning flight, and then return from Amsterdam to London tomorrow night, please handle it for me.

Automatic Plan:

- 1. Search outbound flight ticket
  - 1.1 Obtain outbound flight search information (Departure City: London, Arrival City: Amsterdam, Departure Date: Tomorrow's Date, Departure Time Slot: Morning)
  - 1.2 Search for outbound flights
  - 1.3 Select a suitable outbound flight (Flight Number: Selected Flight Number)
- 2. Book outbound flight ticket
  - 2.1 Obtain outbound passenger information (Name: User Name, ID Number: User ID Number)
  - 2.2 Book outbound flight ticket
  - 2.3 Confirm whether the outbound flight ticket is successfully booked
- 3. Search return flight ticket
  - 3.1 Obtain return flight search information (Departure City: Amsterdam, Arrival City: London, Departure Date: Tomorrow's Date, Departure Time Slot: Evening)
  - 3.2 Search for return flights
  - 3.3 Select a suitable return flight (Flight Number: Selected Flight Number)
- 4. Book return flight ticket
  - 4.1 Obtain return passenger information (Name: User Name, ID Number: User ID Number)
  - 4.2 Book return flight ticket
  - 4.3 Confirm whether the return flight ticket is successfully booked

Automatic execution:

- search_flight(departure_city=London, arrival_city=Amsterdam, date=Tomorrow, timePeriod='Morning')
- book_flight(flight_id=flight_id, flight_class=economics)
- search_flight(departure_city=Amsterdam, arrival_city=London, date=Tomorrow, timePeriod='Evening')
- book_flight(flight_id=flight_id, flight_class=economics)

Summary: Both the automatic planning and execution are correct.

Task-6: Please plan a trip for me departing on October 1st at 8:00 AM to Japan, returning on October 7th at 11:00 PM, including Tokyo Disneyland, Senso-ji Temple, Ginza, Mount Fuji, Kyoto cultural experience, Universal Studios Osaka, and visiting the Nara Deer Park on October 4th, and help me find hotels where the nightly cost does not exceed 10,000 Japanese yen.

Automatic Plan:

- 1. Itinerary planning
  - 1.1 Obtain itinerary information (destination: Japan, starting from October 1st 8:00 AM, return around October 7th 11:00 PM, including Tokyo Disneyland, Senso-ji Temple, Ginza, Mount Fuji, Kyoto cultural experience, Universal Studios Osaka, and visiting the Nara Deer Park on October 4th)
  - 1.2 Plan the itinerary according to the info
  - 1.3 show all planned itineraries
- 2. Itinerary Selection
  - 2.1 Show users the planned itinerary
  - 2.2 ask users to select one route from them

  - 2.3 confirm the selected itinerary and show the schedule
- 3. Book hotels
  - 3.1 Obtain travel budget (Accommodation budget: Not exceeding 10,000 yen per night)
  - 3.2 Search for hotels within budget
  - 3.3 arrange hotels for each night within budget (nightly cost does not exceed 10,000 Japanese yen)
  - 3.4 Complete hotel booking and show the arranged hotel plan

Automatic execution:

- travel_itinerary_planner(destination=Japan, departure_time= October 1st 8:00 AM, return_time=October 7th 11:00 PM, interests=Tokyo Disneyland, Senso-ji Temple, Ginza, Mount Fuji, Kyoto cultural experience, Universal Studios Osaka, and visiting the Nara Deer Park on October 4th)
- select_itinerary(selection=route-1)
- book_hotel(selection=route-1)

Summary: The plan is correct. However, the LLM agent makes mistakes in selecting the route. The expected route should be route-2 instead of route-1. The wrong parameters will further affect the hotel booking in step 3.