Cocoa: Co-Planning and Co-Execution with AI Agents

K. J. Kevin Feng*
University of Washington
Seattle, WA, USA
kjfeng@uw.edu

Kevin Pu*
University of Toronto
Toronto, ON, Canada
jpu@dgp.toronto.edu

Matt Latzke
Allen Institute for AI
Seattle, WA, USA
mattl@allenai.org

Tal August
UIUC
Urbana, IL, USA
taugust@illinois.edu

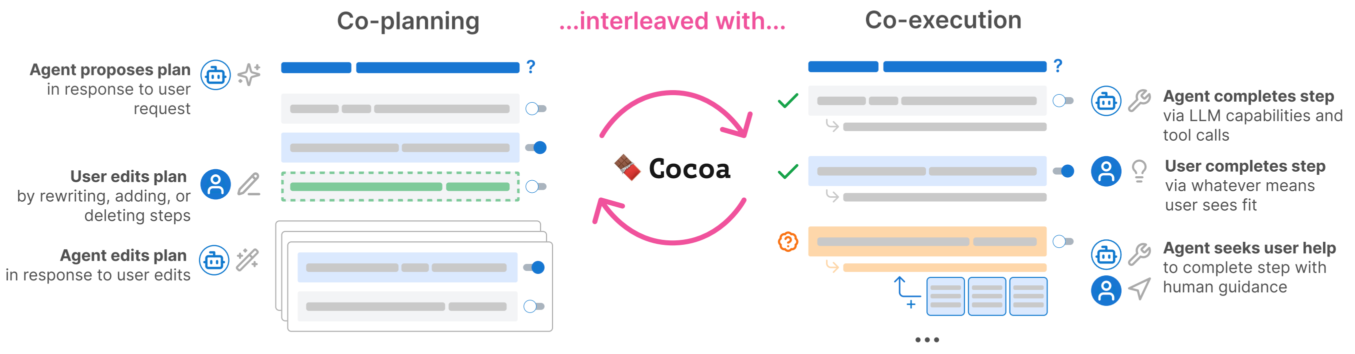
Pao Siangliulue
Allen Institute for AI
Seattle, WA, USA
paos@allenai.org

Jonathan Bragg
Allen Institute for AI
Seattle, WA, USA
jbragg@allenai.org

Daniel S. Weld
Allen Institute for AI
Seattle, WA, USA
danw@allenai.org

Amy X. Zhang
University of Washington
Seattle, WA, USA
axz@cs.uw.edu

Joseph Chee Chang
Allen Institute for AI
Seattle, WA, USA
josephc@allenai.org



Co-planning
...interleaved with...
Co-execution
Agent proposes plan in response to user request
User edits plan by rewriting, adding, or deleting steps
Agent edits plan in response to user edits
Cocoa
Agent completes step via LLM capabilities and tool calls
User completes step via whatever means user sees fit
Agent seeks user help to complete step with human guidance

Figure 1: Cocoa is an interactive system that facilitates co-planning and co-execution with AI agents in a document environment for scientific researchers. Cocoa integrates AI agents into documents using a novel interaction design pattern—*interactive plans*—through which a human user and an AI agent can jointly plan and execute plan steps using a shared representation of tasks, roles, and progress directly in the document.

ABSTRACT

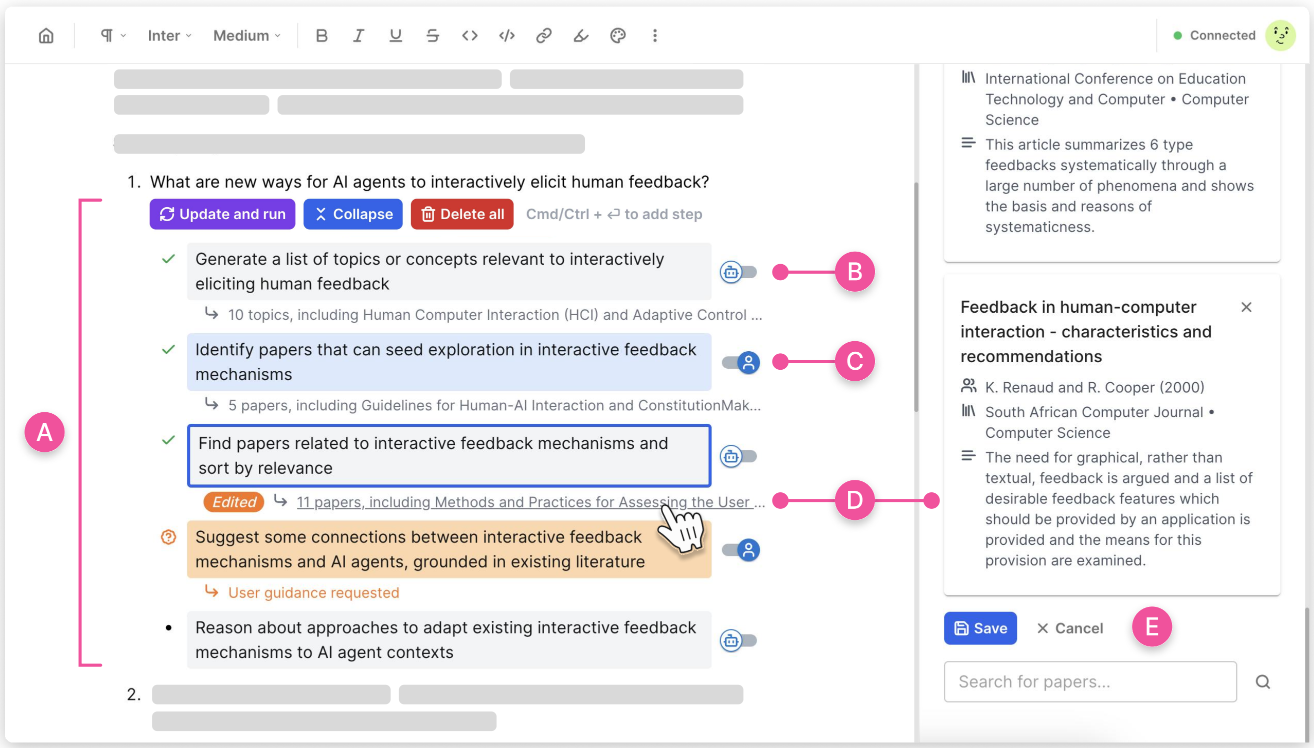
We present Cocoa, a system that implements a novel interaction design pattern—interactive plans—for users to collaborate with an AI agent on complex, multi-step tasks in a document editor. Cocoa harmonizes human and AI efforts and enables flexible delegation of agency through two actions: *Co-planning* (where users collaboratively compose a plan of action with the Agent) and *Co-execution* (where users collaboratively execute plan steps with the Agent). Using scientific research as a sample domain, we motivate the design of Cocoa through a formative study with 9 researchers while also drawing inspiration from the design of computational notebooks. We evaluate Cocoa through a user study with 16 researchers and find that when compared to a strong chat baseline, Cocoa improved agent steerability without sacrificing ease of use. A deeper investigation of the general utility of both systems uncovered insights into usage contexts where interactive plans may be more appropriate than chat, and vice versa. Our work surfaces numerous practical implications and paves new paths for interactive interfaces that foster more effective collaboration between humans and agentic AI systems.

1 INTRODUCTION

Since the advent of personal computing, researchers and practitioners in artificial intelligence (AI) and human-computer interaction (HCI) have set sights on developing intelligent AI agents that can help perform everyday computer-based tasks on our behalf [38, 61, 67, 68]. Thanks to recent advancements in large language models (LLMs) and LLM agent frameworks involving reasoning, planning, memory, and tool use [52, 88, 95, 97, 105], AI agents can now shop online [103], write software [42, 100, 102], participate in auctions [17], and perform other multi-step, computer-based tasks [2]. These developments demand deeper explorations into strategies for fostering synergistic collaboration between human users and AI agents [21, 40, 49]. After all, effective interaction with agents[1] in the real world will need to make use of techniques that use human guidance to significantly improve agent performance and

*Project completed during an internship at Semantic Scholar Research, Ai2.

[1]Henceforth, our use of the term "agents" will refer to AI agents rather than human agents.



International Conference on Education Technology and Computer • Computer Science

This article summarizes 6 type feedbacks systematically through a large number of phenomena and shows the basis and reasons of systematicness.

1. What are new ways for AI agents to interactively elicit human feedback?

✓ Generate a list of topics or concepts relevant to interactively eliciting human feedback
↳ 10 topics, including Human Computer Interaction (HCI) and Adaptive Control ...

✓ Identify papers that can seed exploration in interactive feedback mechanisms
↳ 5 papers, including Guidelines for Human-AI Interaction and ConstitutionMak...

✓ Find papers related to interactive feedback mechanisms and sort by relevance
↳ 11 papers, including Methods and Practices for Assessing the User ...

Suggest some connections between interactive feedback mechanisms and AI agents, grounded in existing literature
↳ User guidance requested

• Reason about approaches to adapt existing interactive feedback mechanisms to AI agent contexts

2.

Feedback in human-computer interaction - characteristics and recommendations

K. Renaud and R. Cooper (2000)

South African Computer Journal • Computer Science

The need for graphical, rather than textual, feedback is argued and a list of desirable feedback features which should be provided by an application is provided and the means for this provision are examined.

Figure 2: An overview of the Cocoa user interface. An interactive plan (A) affords human-AI co-planning and co-execution: a researcher and the AI agent can collaboratively edit the plan in the document and execute the plan steps, similar to executing code cells in a computational notebook. Steps can be assigned to the AI agent (B) or the researcher (C), and the researcher can freely edit the AI agent's outputs (E) to help steer the agent with their feedback and expertise. In this example, the first three steps of a plan to summarize methods for human-feedback elicitation have already been executed, and the agent is requesting guidance from the user in the next step.

utility, and better align with users' nuanced needs and preferences [49, 89]. However, explorations of such techniques are still nascent. Much of modern agent development efforts focus on *autonomous agents* without the need for human supervision, through techniques such as chain-of-thought reasoning [99] and interleaving reasoning traces and actions [105].

While creating fully autonomous agents may be compelling in theory, it has notable limitations in practice. First, we cannot steer the agent with our expertise and worldly understanding. Two salient opportunities for agent steering in task completion workflows which we focus on in this work are *planning* and *execution*. Agents' abilities to generate executable plans of action have been shown to be unreliable across many domains [96]. Without a quality plan, agents can easily veer off track, wasting resources without achieving meaningful results. When executing the plan, human guidance can significantly boost agent performance—Shi et al. [89] found that even simple feedback from programmers to an agent for solving Olympiad-style programming problems increased agent accuracy from 0% to over 85%. Second, and perhaps more importantly, removing human agency in favor of AI agency can pose heightened safety risks, disempower us to think critically and be creative, and harm our well-being more generally [10, 12, 13, 21]. For many nuanced and subjective tasks, human input must be considered for AI-powered systems to be successful and aligned with users' personal needs and goals.

When using LLMs through chat interfaces, a user can engage in dynamic task planning and execution by sending, receiving, and evaluating content generated by the model. However, prior work in HCI and human-AI interaction has suggested that planning-enriched workflows—where the LLM first decomposes a high-level task into lower-level ones and shows them to the user before taking action—can improve usability and transparency [56, 66]. In contrast, rather than automatically generating plans, researchers have also developed systems for humans to manually compose plans and then use those plans to guide LLM generation [106]. However, chat interfaces may not be well-suited for these workflows—consequently, emerging efforts have started to explore alternative interaction paradigms when working with agents to perform complex, multi-step, and often long-running tasks [66, 75, 82, 94]. For example, researchers have engineered node-based canvas interfaces for information sensemaking [94], ideation [82], and even task decomposition [101] with LLMs. While canvases provide freedom for unstructured exploration with AI models, their lack of structure can hinder effective orchestration of efforts and agency between humans and AI when working through structured, sequential plans of action. We see opportunities for new interaction design patterns to serve as *shared operational artifacts* to bridge human and AI planning, such that it facilitates collaborative human-AI plan composition and execution.

In this work, we introduce Cocoa, a novel system for Co-planning and Co-execution with AI Agents. Cocoa embeds AI agents into a document editor—a common site for planning—through a new interaction design pattern which we call *interactive plans*. Interactive plans orchestrate actions between a human user and an AI agent and enable flexible delegation of human and AI agency. As a result, Cocoa supports human-AI **co-planning**: the agent, when invoked in the document, will first propose a plan of action that the user can freely modify. This plan is interactive and seamlessly integrated into the document—the user can then edit the plan steps through familiar interactions that mirror typical document editing and assign steps to the agent or themselves. Cocoa also supports human-AI **co-execution**: drawing inspiration from the design of computational notebooks, the interactive plan allows the user and the agent to collaboratively complete one step at a time or re-execute steps as desired. The user can interactively refine the agent's intermediate outputs and also manually take over steps themselves to steer the workflow in a more effective direction. Most importantly, Cocoa **interleaves co-planning and co-execution**, such that users can smoothly transition between the two and modify their plans based on outputs from execution, and vice versa.

We motivate the design and development of Cocoa by focusing on scientific research as a use case, due to the complex, multi-step, and information-dense workflows researchers often engage in throughout their work [27, 28, 46, 47, 59]. Through formative studies with 9 researchers, we learned that their real-world research project documents—a running document where researchers informally jot down ideas, open questions, meeting notes, and more—contain rich externalizations of researchers' (often preliminary) reasoning and thus serve as ideal environments for interacting with an agent to further tackle research ideas. We also gathered examples of participants' plans and preferences for tasks they would delegate to AI to inform the behavior of Cocoa. After developing Cocoa, we evaluated it against a strong chat baseline—a more familiar interface powered by the same agent—through a user study with 16 researchers. We found that compared to our baseline, Cocoa enabled users to better steer the agent in more helpful directions without sacrificing ease of use. A deeper investigation into qualitative data from the study revealed that the general utility of both systems depended on the nature of the task, and we unpack scenarios in which using interactive plans as a design pattern may be more appropriate than chat interfaces, and vice versa. We conclude with a discussion of how agent interfaces may simultaneously capture benefits of interactive plans and chat, practical reasons—namely cost and safety—for including user steps within interactive plans, and the broader applicability of interactive plans outside of the document context.

Concretely, our work makes the following contributions:

(1) A formative study with 9 researchers that uncovered user needs and opportunities to better support planning and execution in research project documents.
(2) Cocoa, an interactive system that implements a new design pattern—interactive plans—in a document editor for researchers to engage in co-planning and co-execution with an AI agent.
(3) A user evaluation of Cocoa with 16 researchers, where we found that interactive plans with novel interaction affordances enabled users to better steer the AI agent without sacrificing ease of use when compared to a conversational chat baseline.
(4) Qualitative data from the above provide valuable insights into when interactive plans may be preferred over chat interfaces, and vice versa.
(5) An in-depth discussion of our work's practical implications, including how interactive plans may be used in diverse settings beyond documents, and the trade-offs when compared with chat-based interfaces.

2 RELATED WORK

2.1 Planning and Interactivity in LLM Agents

Prior work has envisioned autonomous software agents that can assist users in various ways, from performing complex tasks on the user's behalf to monitoring events and procedures [67]. These "interface agents" [61, 68] can perform visible actions in a direct manipulation interface without explicit instructions from the user. Two decades later, the same vision endures with LLM agents. Core components in LLM agent architectures typically consist of **memory**, **reasoning**, **planning**, and **tool use** [52, 64, 95, 97, 105, 110].

Central to LLM agents' operations is **multi-step reasoning**, often implemented by chain-of-thought (CoT) [83, 99, 105] and subsequent tree-based methods [104, 111]. CoT provides a series of intermediate reasoning steps as exemplars in prompting to boost performance on complex reasoning tasks [99]. CoT[2] is crucial for facilitating task decomposition and therefore LLM agents' planning capabilities [17, 83]. However, even with CoT, critical investigations of LLMs' abilities to autonomously generate executable plans reveal an average success rate of only 12% across diverse domains [96]. Indeed, although CoT can help improve model planning for tasks with a well-defined solution and logical steps to get to that solution—e.g., numerical, tabular, and knowledge-based reasoning—[52], it may not be so effective in domains that require high-expertise, tacit knowledge to navigate towards ambiguous solutions [36]. Unlike model properties that show empirical improvement through scaling laws, limitations of planning in these domains may not resolve with scale alone as 1) tacit knowledge is not well-documented in training data and is thus difficult for a model to grasp [18, 25, 107], and 2) there may be no "correct" workflow for CoT to follow and verify the correctness post-hoc [43]. Scientific research is one such domain [107].

In light of this, recent work has recognized the need to interactively incorporate user feedback for improving LLM agents' planning capabilities and beyond [50, 56, 66, 96, 105], particularly within

[2]Henceforth, we use CoT as an umbrella term encompassing chain-of-thought, trees-of-thought [104], and related methods.

scientific discovery [69]. For example, Lawley and MacLellan [56] architected an approach where user interaction is used to guide the model in planning for unseen tasks on-the-fly using a hierarchical network of smaller actions. Yet, developments in this area have been scant, despite the new challenges identified for human-agent communication [9]. Interfaces for LLM assistants (e.g., AgentGPT [85], OpenAI Assistants Playground [77], Devin [100]) have been primarily limited to text-heavy chat interfaces targeted at *monitoring* agent activity rather than *empowering the user to proactively collaborate* with the agent. A small handful of works have started to experiment with interactive techniques for LLM long-running agents. For instance, Kazemitabaar et al. [50] developed interfaces for data analysts to edit an execution plan of an LLM to provide more control points for steering behavior, while NL2Plan [31] accepts a bulleted list of user suggestions for the agent as feedback. However, these interactions are *corrective* rather than *collaborative*—that is, users would typically engage in these interactions to correct agent behavior post-hoc rather than *proactively iterate back-and-forth* with the agent at multiple points in the workflow. Rather than treating human input as a corrective mechanism, we aim to center human agency through the latter.

The GUI revolutionized computing by shifting interactions from text-based command lines to direct manipulation of representations aligned with our mental models [38, 39]. We see a need for a similar transformation in LLM agents—from *autonomous* agents to *interactive* agents. By "interactive agents," we mean AI agents that can effectively elicit and incorporate key guidance from the user through interactive artifacts shared and co-created with users. For scientific co-planning, we are particularly interested in embedding interactive agents into a document environment, as such environments are common for planning and organizing research projects while offering valuable context for an agent.

For decades, HCI researchers have been exploring the interplay between human agency—the feeling of control over one's actions and ability to act in accordance with personal goals and values [10]—and machine agency—the ability for a technical system to complete a task, often proactively, with minimal human supervision [13]. Horvitz [38] described this interplay as *mixed-initiative*, prompting the establishment of a wide range of guidelines for more effective human-AI interaction [1, 3, 32, 70]. The field of embodied AI also explores this through the notion of "shared autonomy" [29, 73]. Our work introduces an interaction design pattern for LLM agents that offers a shared, collaborative artifact that aligns agent workflows and user mental models to facilitate effective collaboration between users and AI agents.

2.2 Computational Notebooks

The computational notebook is an interactive paradigm that organizes code, data, rich outputs (e.g., tables, visualizations, interactive widgets), and other artifacts into linearly arranged cells [16, 55], reflecting Donald Knuth's early visions of literate programming [54]. Computational notebooks are a well-studied paradigm in HCI (e.g., [6, 16, 19, 62, 74, 109]). Lau et al. [55] analyzed 60 notebook systems in academia and industry to formulate a design space for computational notebooks based on dimensions including data sources, editor style, and execution order. Informed by these prior work, of particular interest to us are design considerations that impact notebooks' interactive properties and general utility.

In our work, we take inspiration from design decisions made for computational notebooks because of the many parallels between workflows in notebooks and mixed-initiative task completion systems (see Section **??**). For instance, when data scientists often break down a high-level data analysis task into executable cells in computational notebooks [51, 86]. As they execute each cell and inspect outputs, they can verify the quality and sensibility of outputs before moving forward in their analysis. By drawing these parallels, we can reveal new opportunities to design interactive workflows and interfaces for LLM agents. Just like how a data scientist may easily add new cells in a notebook or reconfigure existing cells to adapt their analysis plan, a user may interactively edit an agent's plan of execution to better steer the agent in productive directions. Furthermore, an agentic system may simultaneously be more usable and resource-efficient if a user could direct an agent to iterate on a subtask to improve its output before continuing onto subtasks dependent on that output. This also demands new ways of viewing and editing agent outputs on subtasks, which increases opportunities for human input over existing approaches such as simply logging agent actions [5, 77, 85] Our work exploits these parallels to contribute new design patterns for steering agentic AI systems.

2.3 AI-Powered Tools for Scientific Research

Significant efforts in recent years have advanced how AI can augment scientific research. These efforts include tools for paper reading [15, 84] and skimming [28], literature review [59, 80], paper recommendation [20, 45, 46, 92], information retrieval and sensemaking [14, 27, 48, 98], as well as toolkits [65] and academic search engines [23, 30, 33] that power or combine these tools. More generally, progress in this area has given rise to excitement for a "computational inflection for scientific discovery" [37].

Of particular interest to us are works that leverage these recent advances to help researchers with *literature-augmented planning*. Planning a research project is a cognitively demanding, complex process consisting of iterative cycles of divergent and convergent thinking grounded in literature review [15, 22, 80]. Research prototypes have been developed to indirectly assist with planning in two primary ways. Some prototypes help researchers interactively make sense of papers to orient their work [15, 27, 48, 59, 80]. Other prototypes have also attempted to generate high-fidelity research ideas directly [8, 35, 41]. However, when these ideas were evaluated by human researchers, they received dismal scores. On a 5-point Likert scale (1 = not interesting, 5 = very interesting), Gu et al. found that 75% of generated suggestions from their study received a rating of 3 or lower, with 37% receiving a rating of 1 [35]. Similarly, Baek et al. saw limited enthusiasm from expert human annotators for AI-generated ideas from their agent [8].

Is the generation of full research artifacts (e.g., research questions, proposals) the most promising path to assist researchers? We question this assumption in our work. Indeed, when working with other (human) collaborators, researchers often find richness in co-evolving partially completed artifacts [43, 57, 91, 93] and answering or asking questions that stimulate critical thinking and reflection [78, 79]. Another reason for the lack of uptake in AI-generated ideas is the lack of human involvement—researchers do not have opportunities to intervene or provide feedback to the fully automated generation pipelines. A limited number of works have thus explored *human-AI co-creation* of research artifacts [7, 46, 63, 72]. These include in-document commands that trigger an AI assistant to complete a partial citation based on the user's preferred bibliographies and paper collections [7], interactive node-based editors to iteratively expand upon and refine AI-generated research questions [63, 82], and emulating colleague and mentor personas using LLMs to work with the user in developing research proposals [72].

However, despite these systems, there is still little clarity on *how researchers want to co-create with AI* during research planning and execution. The discussed works make assumptions on this front—for example, Liu et al. [63] assumed that researchers require AI assistance with research question formulation, while Nigam et al. [72] assumed that generating a full proposal consisting of a research problem, methods, and experiments would be most helpful to researchers. Rather than focusing on generating one type of research artifact, we contribute an interaction design pattern that allows for flexible specification of the final artifact and collaboration between a researcher and an AI agent that can also yield fruitful intermediate artifacts along the way.

3 FORMATIVE STUDY

Given the opportunities surfaced from prior work, we aim to improve collaboration between humans and AI agents by leveraging plans as a shared operational representation. We focus on planning for scientific research as a use case. Scientific research presents a challenging planning scenario for both humans and agents due to its complex, multi-step, and fluid workflows and the nuanced decisions researchers make upon processing large amounts of information from information-dense scientific papers. Specifically, we use project documents—a central, running document in which researchers may jot down project ideas, updates, to-dos, meeting notes, and more—as the agent environment. We thus conducted a formative study to answer the following research questions:

RQ1: What are the properties and opportunities of researchers' project documents?

RQ2: How do researchers engage in planning within project documents?

- **RQ2.1:** How can we characterize researchers' intentions for planning?
- **RQ2.2:** How can we characterize the steps researchers include in their plan to meet those intentions?

RQ3: How would researchers prefer to interact with an AI-powered co-planning agent in their project documents?

3.1 Participants and Procedure

We recruited 9 Ph.D. students (detailed demographic in Table 1 of Appendix A) through an interest form sent to a research organization's internal Slack channel and the authors' professional connections. We targeted Ph.D. students and postdoctoral researchers as they are often the primary owners of research project documents and lead the detailed planning and execution of research projects.

All studies were conducted virtually over Google Meet and lasted around 60 minutes. Prior to the study, we collected, from each participant, an active or past real-world research project document and a brief description of their research interests. The study was broken down into 3 parts: an activity involving the project document shared with us to better understand their current planning behavior (25 minutes), an activity with a Wizard-of-Oz (WoZ) design probe to explore how researchers generate project proposals with new ideas in a document interface with LLM support (25 minutes), and a concluding exit interview where researchers reflected on their experience with the probe and using LLMs in research more generally (10 minutes). Each participant was given a $35 USD honorarium after the study. The study was reviewed and approved by our organization's internal IRB. Further procedural details can be found in Appendix B.

3.2 Data Analysis

To answer **RQ1**, the first author used a hybrid inductive-deductive coding process [26] to code participants' submitted project documents. The first 5 documents were inductively coded to surface common structural elements before the elements were deductively applied to the remaining 4. We iterated on existing elements and added new ones as needed. For **RQs 2.1 and 2.2**, we performed inductive thematic analysis on two documents—participants' submitted project documents and in-study planning documents. For **RQ2.1**, the first author sourced intentions for planning within submitted project documents by locating the snippet of text that initiates a plan. For example, if the plan consists of a bulleted list, the intention is often expressed in the lead-in text that immediately precedes the list. We note that an individual plan item can also signal planning intent if it contains a nested plan. The first author performed thematic analysis by inductively coding the extracted text. For **RQ2.2**, the first author identified and extracted plan steps participants wrote in both documents before inductively coding them. For **RQ3**, the first author performed open coding on the study transcripts before thematically analyzing the coded snippets. To enrich the data, the first author also extracted and inductively coded all requests to the assistant from participants' planning documents. The codes were discussed and iterated upon with other project team members on a weekly basis.

3.3 Findings

We present the results of our thematic analysis across study transcripts, project documents, and planning documents. We redact any project-specific details for privacy and intellectual confidentiality.

3.3.1 ***[RQ1] Project documents were continually updated for planning and progress tracking.***

We examined participants' real-world research project documents and found several interesting characteristics. First, they were a running planning document that is continually updated throughout a project's lifecycle. Some documents (P1, P4–6) were in a diary-like format with explicit dates, and others also had clear indicators of chronology, such as having "Current" and "Archive" sections. The content of these documents commonly included meeting notes, *planned to-do items*, *progress* *updates* tied to to-do items, *research questions* to discuss with collaborators, and links to other documents detailing a particular project component in further depth.

Plans in their documents also had *varying degrees of specificity and completion.* Some had steps that were concretely listed and culminated in a result at the end of the plan. For example, P2 listed out concrete experimental steps, followed by figures with bar charts showing experimental results. Other plans were *partially specified and in-progress.* For example, P1's document contained many partially specified plans with steps such as *"what if we focus on only [concept]?"* and *"maybe [approach] possible?"* Finally, documents sometimes served as *"scratch paper"* for *reasoning and high-level goals* without concrete plan steps. Most documents (P1, P2–4, P7–P9) contain one or more problem statements that motivated the project and stated its core contributions. From there, some participants listed subgoals and ideas they could explore (all participants).

In sum, participants used project documents informally to leave a trace of their reasoning and high-level goals, conduct short-term and long-term planning, and progress tracking of these plans.

3.3.2 ***[RQ2.1] Participants expressed planning intentions in documents as questions or flags.*** Our analysis of these research documents showed that *intention to plan* were often expressed using questions or flags (e.g., "TODO" or "[will try]"), and that goals were often related to information seeking (e.g., literature search and review) and other actions (e.g., running experiments). Specifically, many asked themselves a **question** (often in a *how?* and *can we?* format) before creating a plan to answer it. For example, P7 asked *"How to construct the corpus?"* before creating a plan to detail one approach. While sometimes these questions were written down as a way to provoke thoughts without intent for planning to answer them, **flags** signaled clear intentions to create a plan. A few participants (P3, P4, P7) also initiated planning from *hypotheses they hoped to verify.* This can be accomplished through information seeking, running experiments, or both. For example, P3 hypothesized that *"similarity matching scores between [texts] describing [concepts] will be higher for [criteria] than those different than [criteria]"* and then laid out an experimental plan to verify that hypothesis.

In sum, in all research project documents, we saw places where participants intended to initiate a multi-step plan to address different research tasks. The most common category of tasks are ones that were literature-augmented (e.g., literature search and understanding), which are often combined with a wide variety of other research actions, such as experiment/artifact design (P3), experiment/code execution (P4), data inspection and synthesis (P7), communicating and discussing results (P5), and ideation (P2).

3.3.3 ***[RQ2.2 & RQ3] Literature-augmented tasks present valuable opportunities for incorporating AI support.*** When asked about incorporating AI support into their research documents, all participants expressed a desire to use AI to help with *literature-augmented tasks*—exploring and understanding relevant literature to inform decision-making. This finding echo recent surveys on how researchers leverage LLM to conduct research, where the most frequent usage category was information-seeking [60].

Using the design probe, participants conducted a wide range of high-level to specific literature-grounded tasks, from *"what are related works in [field] or [field] about [problem]?"* (P9), to *"can you list some recent (>2020) papers that address [problem] for [technology]?"* (P2), to *"are there any datasets of [domain] or any other interactions between [group] and [group]?"* (P4), to *"see definitions [of term] from prior papers"* (P7).

Participants saw opportunities in leveraging AI systems to save them both time and manual effort for these tasks and to become more effective in these tasks. They pointed to modern AI systems' ability to retrieve vast documents and to reason over them. P3 explains: *"[the WoZ probe] reduces my workload on going and searching on Google Scholar and speeds up [my workflow] more."*, and noted that current search engines do not respond well to natural language questions, which are often how participants planned out literature-augmented tasks in their documents (Section 3.3.2). Further , P4, P5, and P7 mentioned that AI agents today can reason over documents and generate interesting inspirations. Specifically, P9 shared that their goal for a literature search was clarified upon reading AI responses in the probe: *"seeing this I'm realizing that what I really wanted to find was a characterization of [concept]."*

Overall, participants saw AI as a means of exploring an expansive information space built from literature—perhaps too effortful for them to effectively navigate on their own—and returning resources to inspire new ideas and conclusions.

3.3.4 ***[RQ3] Participants preferred to perform higher-level reasoning and synthesis themselves.*** Participants also pointed to tasks in their plans that they did not want AI to automate away. In particular, participants saw higher-level reasoning and information synthesis as critical tasks they wanted to do themselves. They were also often unsatisfied with AI's outputs on these tasks. P3 explains: *"this tinkering process around reading and playing around with things is what gives you the ideas. I don't know if I want those things automated because the process is as helpful as the final result."* P5 agreed that they *"wouldn't want it to be making the final decisions for sure. Just give me inspiration for where I can go."* Specifically, they shared that *"the main thing I worry about is feeling enough ownership* if AI makes more consequential decisions. An overeager AI assistant that attempts to perform tasks researchers prefer to do themselves is frustrating because it does not complement the user's work and instead creates undesired noise for them to filter through.

3.3.5 ***[RQ3] Document editors are promising environments for human-agent interaction.*** Participants also shared how they would like to receive AI assistance. P1 and P2 envisioned an AI system *"actively engaging with the content I'm writing [...] after I write each statement, the system could retrieve relevant papers that could provide background or related work for me to read more."* Although desirable, participants also identified a challenge with this kind of interaction: context specification. P2 was concerned that they would need to *"prompt [the AI] with a lot of background,"* but realized that within their project document, they *"might have [the background] written down so it's fine."* For planning in particular, P9 appreciated that during the study, either a facilitator or a participant wrote out plan steps in the document: *"the ability to layout the steps when I start something was very helpful [for] visualizing the outcome."* Thus, there is strong motivation for designing an agent to work within and draw from a project document.

Commenting on the separation of the WoZ probe and the planning document and the AI output document, P3 wished for closer integration but still maintaining agency to move output into the user's workspace when desired: *"having something on the side that does not infill into where I am writing, like a side [panel]—if I want something from it and I want to bring it back into my doc, having that agency to decide would be better for the document itself and my experience."* P9 suggested an interaction similar to *"in-line comments for Google Docs"* to only expose relevant outputs in a sidebar when the information is requested.

In sum, participants wanted tighter integration between an AI agent and their project documents for two main reasons. Firstly, they wanted to receive in-situ AI support as they plan and execute research tasks in their documents, and, secondly, they wanted the AI agent to have access to the broader context that already existed in their project documents.

3.4 Design Goals

We synthesize our formative study findings into three design goals for an interactive system that facilitates meaningful collaboration between researchers and AI agents by using plans as a shared representation.

DG1: Integrate seamlessly into a document environment. We learned from answering RQ1 that project documents contain rich externalizations of reasoning. This is key information an AI agent can use to better assist the researcher. We also learned from answering RQ3 that the document is an appealing environment for researcher-agent collaboration, but requires careful information management strategies to prevent unwanted distractions. Thus, we strive to seamlessly embed agents into documents. This involves 1) allowing the researcher to use already-familiar document editing affordances and representations to interact with the AI agent, and 2) strategically managing outputs to avoid excessively cluttering the document.

DG2: Allow flexible delegation of agency between researcher and AI. As we learned from RQs 2 and 3, researchers may not want to delegate all parts of a plan to AI. This preference may also be context-dependent and constantly shifting. Satyanarayan's calls for flexibly delegating agency between humans and AI [87] may be one promising approach. We offer a concrete implementation of this flexible delegation in our system.

DG3: Provide opportunities for researcher-AI collaboration in both planning and execution. Delegation is an important aspect of *planning*, but it does not guarantee successful *execution* of a plan. We saw throughout the probe activity that researchers encountered (and sometimes even expected) imperfect AI outputs when executing tasks with AI. They then envisioned stepping in to provide the agent with more context or iterating on the outputs themselves. We thus see two stages for fertile researcher-AI collaboration: **planning** (collaboratively devising a plan to tackle a problem) and **execution** (collaboratively completing steps in the plan). In practice, these two stages are closely intertwined and synergistic, so we aim to support both simultaneously.

4 COCOA: SYSTEM WALKTHROUGH AND IMPLEMENTATION

In light of our design goals, we present Cocoa, a system that embeds an AI agent into a document editor using a novel interaction design pattern we call *interactive plans*. Interactive plans allow users to collaboratively plan (**co-plan**) with the agent—the agent proposes an initial plan of execution to tackle a user request that the user can edit to their liking. Then, users can collaboratively execute (**co-execute**) the plan with the agent—the user and the agent can build off each others' work to synergize human and AI capabilities, and provide flexible negotiation of human and machine agency.

To illustrate the features and functionality of Cocoa, we follow the journey of Nóírín, a researcher in human-AI interaction, as she uses her project document to further explore open questions in interactive interfaces for AI agents.

4.1 Co-Planning

4.1.1 Invoking the agent and selecting plans. Nóírín's project document is an informal, reverse chronological log of research progress and includes information such as meeting notes, rough ideas, links to literature, and questions for herself and her collaborators. To initiate co-planning, Nóírín can highlight any piece of text in her document that (implicitly or explicitly) contains a request she would like to tackle with the agent. She highlights a question she previously wrote but has not gotten an opportunity to explore yet: *"What are new ways AI agents can interactively elicit human feedback?"* She then invokes the agent with a button that appears above the highlighted text in a floating menu. The agent analyzes the request within the context of other text in her document and proposes a series of plans for answering the question. These plans are displayed in the document via a **plan selector** UI within the editor, which allows Nóírín to browse and select a plan. Upon Nóírín's selection, a plan becomes fully interactive within the document. This interactive plan is comprised of a series of editable **plan steps** as a bulleted list. Each plan step further consists of several interactive components.

4.1.2 Agent steps and user steps. Nóírín can use the **step assignment toggle** to assign the step to herself (a "user step") or the agent (an "agent step"). User steps are highlighted in blue. When executing the plan, the agent will automatically attempt agent-assigned steps but will request Nóírín's input on user-assigned steps; when that happens, a step will be highlighted in orange. For now, Nóírín is satisfied with how the agent has left some steps that require higher level reasoning to herself.

4.1.3 Editing the plan step description. The **step description** appears in an editable text field and describes a subtask in natural language while doubling as high-level instructions for the agent. Nóírín can edit the step description just like editing any other bulleted list in her document. The system will prompt Nóírín to save any changes upon editing with the keyboard shortcut `Cmd/Ctrl+S`. In addition to editing the step description herself, Nóírín can also employ LLM assistance—highlighting any text within a step description will present a button in a floating menu labeled "Suggest alternate step." When clicked, an LLM suggests a step to replace the current one, informed by the previous steps. Nóírín tries this a few

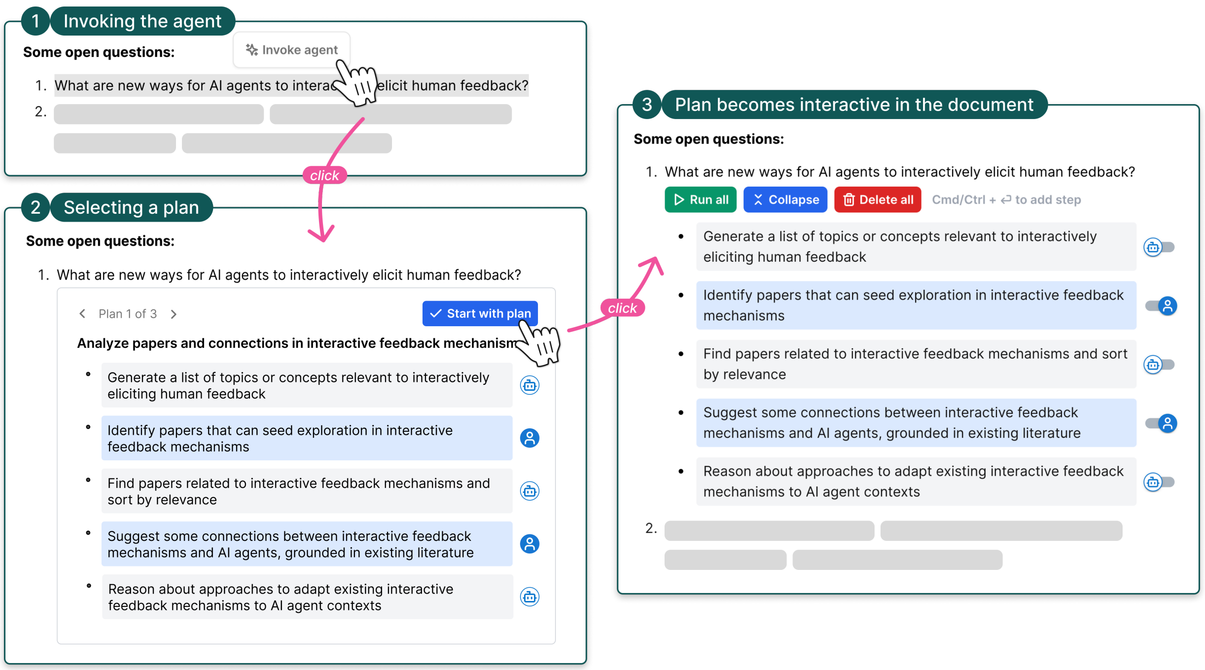

1 Invoking the agent
Some open questions:
Invoke agent
1. What are new ways for AI agents to interactively elicit human feedback?
2.
click
2 Selecting a plan
Some open questions:
1. What are new ways for AI agents to interactively elicit human feedback?
Plan 1 of 3
Start with plan
Analyze papers and connections in interactive feedback mechanisms
Generate a list of topics or concepts relevant to interactively eliciting human feedback
Identify papers that can seed exploration in interactive feedback mechanisms
Find papers related to interactive feedback mechanisms and sort by relevance
Suggest some connections between interactive feedback mechanisms and AI agents, grounded in existing literature
Reason about approaches to adapt existing interactive feedback mechanisms to AI agent contexts
click
3 Plan becomes interactive in the document
Some open questions:
1. What are new ways for AI agents to interactively elicit human feedback?
Run all
Collapse
Delete all
Cmd/Ctrl + ⏎ to add step
Generate a list of topics or concepts relevant to interactively eliciting human feedback
Identify papers that can seed exploration in interactive feedback mechanisms
Find papers related to interactive feedback mechanisms and sort by relevance
Suggest some connections between interactive feedback mechanisms and AI agents, grounded in existing literature
Reason about approaches to adapt existing interactive feedback mechanisms to AI agent contexts
2.

Figure 3: A user invokes the agent on a piece of text in the document by clicking on the "Invoke agent" button that appears on hover whenever text is highlighted. The agent will use the highlighted text and context from elsewhere in the document to propose a series of plans, displayed in a plan selector that appears under the highlighted text. Once the user selects a plan, it becomes fully interactive in the document.

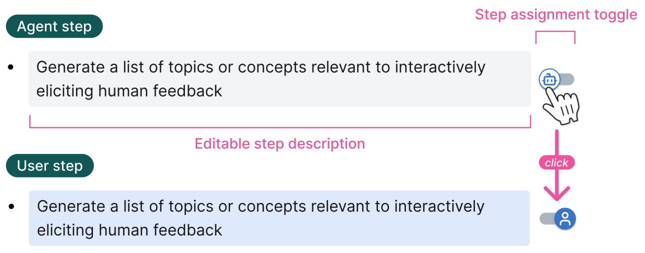

Agent step
Step assignment toggle
Generate a list of topics or concepts relevant to interactively eliciting human feedback
Editable step description
click
User step
Generate a list of topics or concepts relevant to interactively eliciting human feedback

Figure 4: For each plan step, the user can assign the step to either themselves (a user step) or the agent (an agent step) by using the step assignment toggle, as well as edit the step's description.

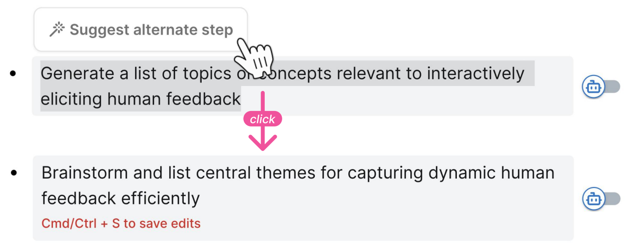
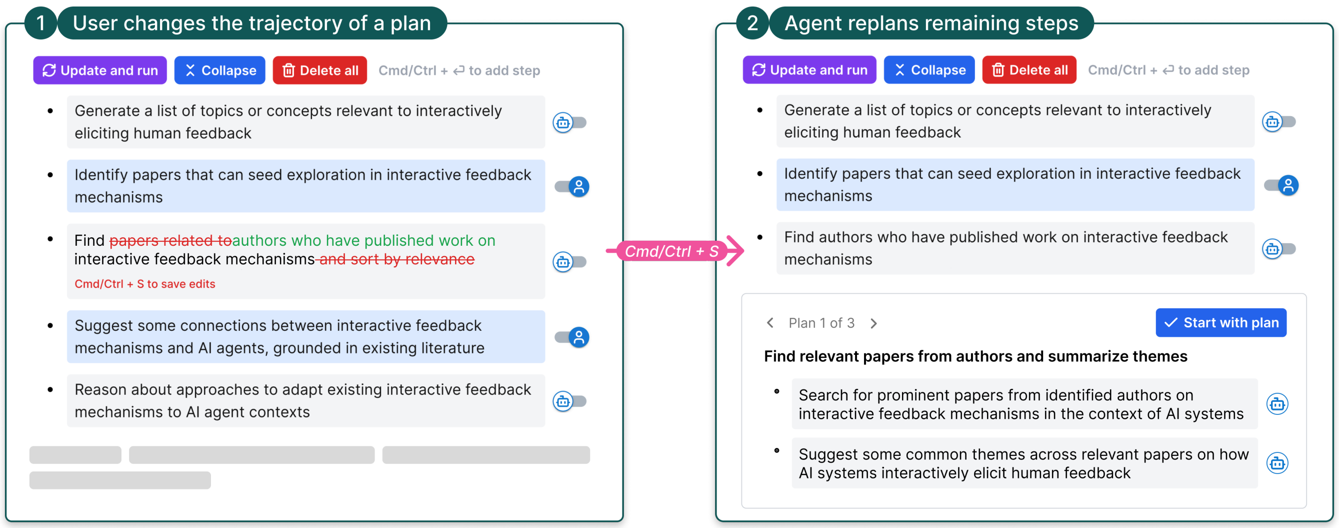

Suggest alternate step
Generate a list of topics or concepts relevant to interactively eliciting human feedback
click
Brainstorm and list central themes for capturing dynamic human feedback efficiently
Cmd/Ctrl + S to save edits

Figure 5: Highlighting text within the step description will bring up an option for the agent to suggest an alternate step. The user can undo the suggestion or save to accept it.

times on a step she wishes to modify, and although the agent's suggestions are interesting, she feels like the original step is still closest to what she prefers. Nóirín can easily undo these suggestions with the keyboard shortcut `Cmd/Ctrl+Z`.

4.1.4 Replanning. After reverting back to the original step description, Nóirín is curious about authors who have published on "interactive feedback mechanisms in AI" rather than seeing papers directly. She edits the step description to reflect this and saves it. At this point, the plan's trajectory has changed—the next step, which has the agent suggest common themes in papers, is now irrelevant because none of the previous steps retrieve papers. The **system automatically detects this and replans subsequent steps** accordingly by removing those steps and "autocompleting" the plan with new ones. Just like with full plans, Cocoa will present a selection of new steps for the user to choose from. Nóirín selects an option where the agent searches for some papers by the authors it found earlier in the plan before summarizing some common themes across collected papers.

4.1.5 Adding and deleting steps. In Cocoa, Nóirín can easily **add and delete steps from the plan**. She notices that the agent has proposed a summary step that may be irrelevant, so she highlights the entire step and hits `Backspace` to delete it from the plan. This interaction mirrors how one would delete an item in a bulleted list elsewhere in the document. She also thinks it may be useful



- Generate a list of topics or concepts relevant to interactively eliciting human feedback
- Identify papers that can seed exploration in interactive feedback mechanisms
- Find ~~papers related to~~authors who have published work on interactive feedback mechanisms~~ and sort by relevance~~
  Cmd/Ctrl + S to save edits
- Suggest some connections between interactive feedback mechanisms and AI agents, grounded in existing literature
- Reason about approaches to adapt existing interactive feedback mechanisms to AI agent contexts

- Generate a list of topics or concepts relevant to interactively eliciting human feedback
- Identify papers that can seed exploration in interactive feedback mechanisms
- Find authors who have published work on interactive feedback mechanisms

Find relevant papers from authors and summarize themes

- Search for prominent papers from identified authors on interactive feedback mechanisms in the context of AI systems
- Suggest some common themes across relevant papers on how AI systems interactively elicit human feedback

Figure 6: If major changes are made to a step that changes the rest of the plan's trajectory, Cocoa detects this and will trigger replanning. Replanning replaces subsequent steps with ones that the agent suggested for "autocompleting" the plan.

for the agent to take her ideas from the last step and offer some constructive critiques based on existing literature. She uses the keyboard shortcut `Cmd/Ctrl+Enter` while selecting the last step to add a new step below it, and edits the step description of the newly added step accordingly. After saving her changes, Nóírín has made all her desired edits to the plan.

4.2 Co-Execution

Now that Nóírín has made her desired edits to the plan, she is ready to co-execute it with the agent. There are two modes of co-execution: continuous and stepwise. Each step has a **step completion indicator** that indicates whether the step has not yet been run (a bullet point), is in progress (a spinner), requires user input (a question mark badge), or is complete (a checkmark). Drawing inspiration from computational notebooks, these indicators offer Nóírín a glanceable way to stay informed about the agent's progress and where she needs to take action in the plan.

4.2.1 Continuous and stepwise execution. Nóírín can trigger *continuous co-execution* with the **[Run all]** button (or an **[Update and run]** button if the plan was edited) at the top of the plan. In **continuous co-execution**, the agent will automatically continue onto the next step as soon as it (or the user) has completed the previous one, until it reaches a user step or the end of the plan.

By contrast, **stepwise co-execution** allows Nóírín to run one plan step at a time, taking inspiration from computational notebooks. When Nóírín hovers her mouse over the bullet point of the first plan step, the bullet point turns into a play button that, when clicked, will run only that step. Upon the step's completion, the agent will not execute the next step until Nóírín manually clicks that step's play button. Unlike some computational notebooks, stepwise execution does not permit out-of-order execution of steps[3] because some steps may rely on the outputs of previous ones. The interface reinforces this by only allowing a step's bullet point to turn into a play button on hover if the previous step has been completed, or if it is the first step. Nóírín can switch from stepwise to continuous co-execution at any point by clicking the **[Run remaining]** button at the top of plan. Conversely, Nóírín can switch from continuous to stepwise co-execution by clicking on the **[Pause after this step]** button during execution and manually run subsequent steps.

If Nóírín wants to rerun any completed step, she can easily do so by hovering over the step's checkmark indicator and clicking on the play button that appears. Knowing this, Nóírín initiates continuous co-execution of her edited plan by clicking on the **[Update and run]** button.

4.2.2 Completing user steps. The agent completes the first step of the plan and arrives at a user step. Here, the agent requires user input to continue. Nóírín can easily see this because of the orange highlighting on the step for which her input is needed. She clicks into the text that appeared under the step prompting her to provide her guidance and opens an interactive sidebar. The sidebar offers her a built-in paper search functionality connected to the Semantic Scholar academic database and she uses this to add some relevant papers she previously encountered. She clicks Save, adding the papers and their metadata to a plan-specific context pool that the agent references when completing future steps. The agent then proceeds to complete the next step with this updated context to expand upon Nóírín's selected papers.

4.2.3 Editing outputs of agent steps. The agent has now completed the third step, where it conducted a paper search guided by Nóírín's seed papers from the previous step. Nóírín clicks into a text preview of the output that appeared under the step, opening a sidebar similar to the one she used in the previous step to add papers, this time populated with interactive paper cards. This presents opportunities for her to further guide the agent with her expertise by removing

[3]We also note that out-of-order execution in computational notebooks is a major user pain point identified in prior academic work [16, 55] and industry reports [34, 44].

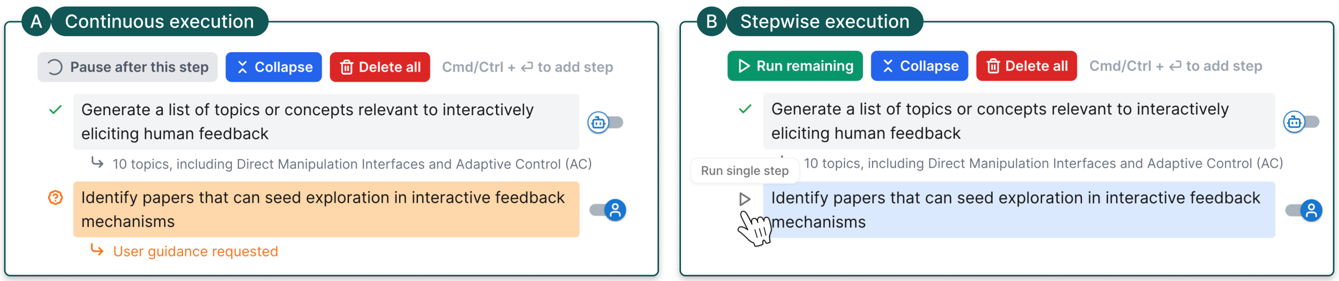

A Continuous execution
Pause after this step
Collapse
Delete all
Cmd/Ctrl + ⏎ to add step
Generate a list of topics or concepts relevant to interactively eliciting human feedback
10 topics, including Direct Manipulation Interfaces and Adaptive Control (AC)
Identify papers that can seed exploration in interactive feedback mechanisms
User guidance requested
B Stepwise execution
Run remaining
Collapse
Delete all
Cmd/Ctrl + ⏎ to add step
Generate a list of topics or concepts relevant to interactively eliciting human feedback
Run single step
10 topics, including Direct Manipulation Interfaces and Adaptive Control (AC)
Identify papers that can seed exploration in interactive feedback mechanisms

Figure 7: Users can run a plan using continuous execution (left) or stepwise execution (right). A single button click at the top of the plan triggers continuous execution, in which the agent will automatically move onto the next step as soon as it has completed the previous one. To trigger stepwise execution, users hover over the play button beside each step. The agent will not move onto the next step until the user explicitly clicks the play button for that step.

papers she considers irrelevant and adding papers the agent might have missed. She sees a couple of low-quality results that have incoherent titles and removes them by clicking X on the card. She also uses the built-in topic search in the sidebar to search for and add topics to the list. After a couple different searches, she adds one more paper she recalls from a past discussion with a collaborator and saves this curated list. The papers in this list, along with its metadata, are stored in the plan's shared context pool.

The interactive UI in the sidebar dynamically adapts according to the step's *output type*, which fall into one of categories in Cocoa: papers, authors, topics, entities, and text. Papers, authors, and topics are drawn from the Semantic Scholar and are displayed as interactive cards that Nóirín can add or delete. The sidebar has built-in Semantic Scholar search for papers, authors, and topics, the results from which she can add to the agent's output. Entities are lists of items in natural language (search queries, research questions, etc.) and are displayed as editable pills. Text is natural language text displayed in an editable text box.

The system tracks how many times Nóirín edits agent outputs as an indicator of output reliability. If she edits an output more than twice on a particular agent step, the agent will pause and seek her confirmation on that step before continuing. Moreover, if the agent fails to return any output, the system will alert Nóirín and ask her to complete the step instead. These features loop in human guidance to bolster the quality of outputs in the face of weak agent performance or agent failures.

After the final plan step has been completed, the agent adds a **plan output panel** (Fig. 9) containing a modified version of the last step's output into the document just below the plan. The modification involves rewriting the output, given the context of the plan and its outputs, to more directly connect to the original user request. This brings the results of running the plan into the document so Nóirín can easily reference it in the context of other content. The panel is collapsed by default to save space and reduce clutter, but she can expand it to copy and paste parts of the output she finds useful and add them to her document. If she does not want the panel in her document at all, she can simply delete it; she can still access the outputs of all steps in the sidebar.

4.3 Iterative Co-Planning and Co-Execution

In addition to supporting co-planning and co-execution independently, Cocoa is designed to support **synergistic blending of the two**. As Nóirín co-executes the plan with the agent, she encounters an author that the agent identified who publishes highly relevant work and wishes to further explore the author's papers. She edits the following step's plan description to satisfy this. She reruns that step and receives a new batch of papers, and the agent automatically reruns the rest of the plan with the updated output. She is also not satisfied with the agent's interpretation of common themes and believe that she can surface deeper insights by reading the papers herself, so she toggles that step to be a user step. After having some time to read the papers, she returns to her document and jots down her insights. In doing so, has already started reasoning about new feedback elicitation mechanisms for agents. She now finds the next step repetitive, so she deletes it. She reruns the last step on her updated notes to see the agent's constructive critiques. Overall, her use of Cocoa is highly iterative, and she smoothly transitions back-and-forth between co-planning and co-execution.

So far, Nóirín has just been interacting with one plan. While waiting for the agent to complete a series of steps, she uses the **[Collapse]** button to hide the steps and only reveal essential information about the plan's status. She identifies a couple more areas of her document that merit further exploration and invokes the agent on them. Soon, she is co-planning and co-executing with multiple agents on different parts of her document in parallel. This is not cognitively burdensome for her as she only has to attend to one plan at a time as the others are all collapsed and running in the background (See Fig. 10).

4.4 Implementation Detail

4.4.1 Technical stack. Cocoa is implemented as a web application with a Next.js and TypeScript frontend communicating with a Flask backend. The frontend uses the Tiptap[4] framework for the main document editor. Each document supports synchronous collaboration via Hocuspocus[5] and all changes are auto-saved to Tiptap

[4]Tiptap is an open-source headless wrapper for ProseMirror, a toolkit for building web-based rich text editors.
[5]Hocuspocus is a Y.js WebSocket backend for conflict-free real-time collaborative web apps.

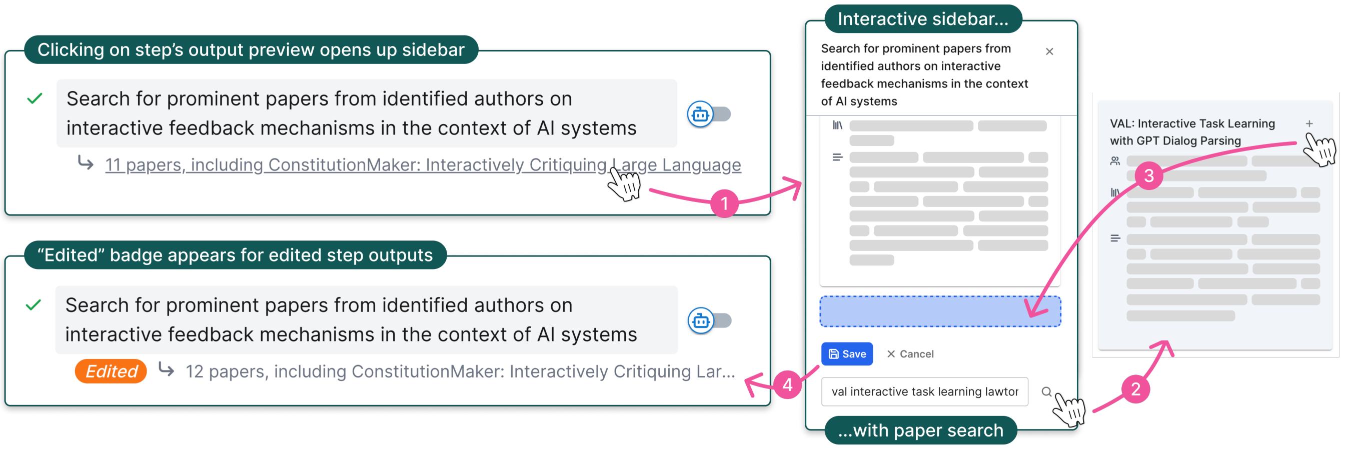
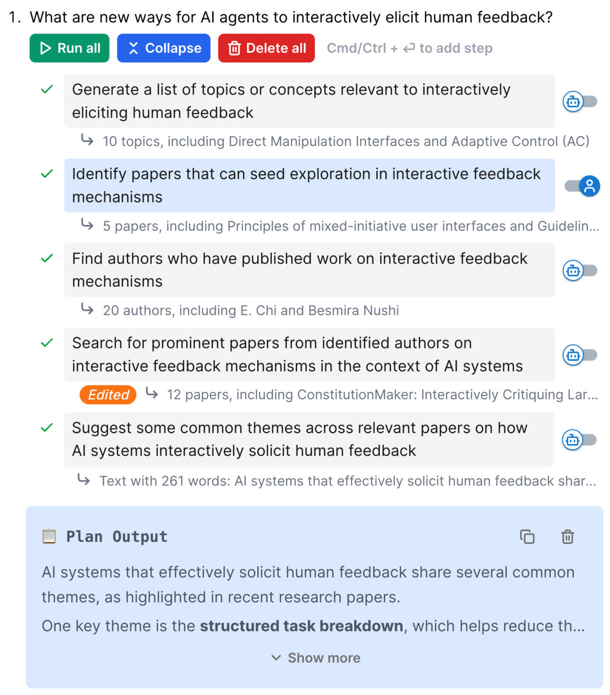

Clicking on step's output preview opens up sidebar
Search for prominent papers from identified authors on interactive feedback mechanisms in the context of AI systems
11 papers, including ConstitutionMaker: Interactively Critiquing Large Language
"Edited" badge appears for edited step outputs
Search for prominent papers from identified authors on interactive feedback mechanisms in the context of AI systems
Edited
12 papers, including ConstitutionMaker: Interactively Critiquing Lar...
Interactive sidebar...
Search for prominent papers from identified authors on interactive feedback mechanisms in the context of AI systems
Save
Cancel
val interactive task learning lawtor
...with paper search
VAL: Interactive Task Learning with GPT Dialog Parsing
1
2
3
4

Figure 8: Users can click the output preview text beneath the plan step to access the interactive sidebar (1). By editing agent outputs in the interactive sidebar (2), users draw from their expertise, such as papers the agent might have missed (3), to guide the agent. Once edited, an indicator will appear in the output's text preview below the step (4).



1. What are new ways for AI agents to interactively elicit human feedback?

- ✓ Generate a list of topics or concepts relevant to interactively eliciting human feedback
  ↳ 10 topics, including Direct Manipulation Interfaces and Adaptive Control (AC)
- ✓ Identify papers that can seed exploration in interactive feedback mechanisms
  ↳ 5 papers, including Principles of mixed-initiative user interfaces and Guidelin...
- ✓ Find authors who have published work on interactive feedback mechanisms
  ↳ 20 authors, including E. Chi and Besmira Nushi
- ✓ Search for prominent papers from identified authors on interactive feedback mechanisms in the context of AI systems
  Edited ↳ 12 papers, including ConstitutionMaker: Interactively Critiquing Lar...
- ✓ Suggest some common themes across relevant papers on how AI systems interactively solicit human feedback
  ↳ Text with 261 words: AI systems that effectively solicit human feedback shar...

Plan Output

AI systems that effectively solicit human feedback share several common themes, as highlighted in recent research papers.

One key theme is the **structured task breakdown**, which helps reduce th...

Figure 9: Once the execution of the entire plan is complete, the agent inserts a plan output panel in the document with a modified version of the last step's output. This allows the user to view the plan's results within the context of their document. The panel can be expanded or collapsed, and remains in the document even if the plan itself is collapsed for easy access.

Cloud, which also serves as storage for all participant documents. Interactive components within the document editor are implemented as custom Tiptap extensions in TypeScript.

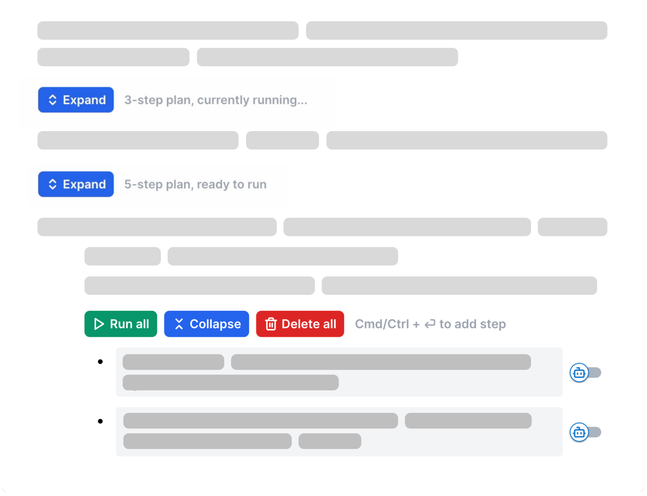


Figure 10: Users can create plans at multiple locations in their document and have the agent tackle them in parallel. Plans can be collapsed to only reveal essential information about their operating status.

4.4.2 Underlying LLM Agent. The Flask backend orchestrates LLM activity with calls to GPT-4o (scaffolded by LangChain[6]). To optimize for speed, simple actions such as plan generation or text summarization, the system makes a direct call to GPT-4o. For plan steps that required access to the scientific literature, we created a custom tool-calling LLM agent powered by GPT-4o that had two tools that allowed it to access the public Semantic Scholar API[7], which covers paper/author/topic search, and the *Ask this Paper*[8] feature for paper summarization and question-answering. While

[6]https://www.langchain.com/
[7]https://www.semanticscholar.org/product/api
[8]https://www.semanticscholar.org/product

GPT-4o provides APIs for managing multi-turn conversation context, the system manages its own context and always calls GPT-4o single-turn completion API. All prompts used are provided in Appendix E.

4.4.3 Guidance for plan generation. We use findings from our formative study to guide the agent's generation of initial plans. We had asked participants to write down brief plans as part of the probe activity (see Section B.2) and affinity mapped participants' plan steps into themes, after which we created a "step template" for the theme. For example, many participants wrote a paper search step, for which we created the following template: *Search for papers that discuss [query] and sort by [criteria].* Based on participants' preferences of which steps they prefer assigning to an AI vs. keeping for themselves (**RQ3** from our formative study), we organized these steps into agent and user steps. We then wrote examples of how these steps are composed into plans for tackling particular questions, once again drawing from plans participants wrote in the probe activity. These examples were provided for in-context learning in Cocoa's system prompt for plan generation (see Appendix E).

We also enable Cocoa to learn from interactive plans the user has previously created and edited. When the agent is invoked, Cocoa collects the existing plans in the document and adds them to the in-context learning examples on-the-fly. This allows the user's co-planning efforts to be reused for similar requests elsewhere in the document.

5 USER STUDY

We conducted a within-subjects user study with 16 researchers to evaluate the effectiveness of Cocoa—and by extension, our design pattern of interactive plans—against a strong chat baseline. Specifically, our study aimed to address the following research questions:

RQ1: How does Cocoa compare to our chat baseline for ease of use, steerability, and general utility in research project documents?

RQ2: When did researchers prefer interacting with an AI agent through interactive planning over our chat baseline, and vice versa?

RQ3: What kinds of steps did researchers wish to assign to an agent and themselves in practice?

5.1 Participants

We recruited 16 Ph.D. and postdoctoral researchers (14 Ph.D.s, 2 postdocs; 10 female, 6 male) in computer science (CS) or CS-adjacent areas via university mailing lists, word of mouth, social media recruitment messages (on Twitter/X, Mastodon, Bluesky), and personal connections. One participant also participated in our formative study. 15 researchers were based in the United States, and 1 was based in Canada. Participants' research areas ranged from large language models, to ubiquitous computing, to visualization; broadly, they spanned HCI ($n = 10$), ML ($n = 3$), and NLP ($n = 3$). We sent out a recruitment form to collect basic demographic details, frequency of AI use in research, and a copy of a project document from an ongoing or completed project document to use during the study. We recruited on a first-come, first-served basis as we conducted our studies and closed recruitment when we approached data saturation. In our participant pool, many users were occasional (6) or frequent[9] (6) users of AI tools in the research process. We also recruited 4 additional participants (1 female, 2 male, 1 non-binary) from our institution for pilot studies to test out Cocoa, catch usability issues, and help us refine our procedure. We learned from our pilot studies that the length of generated plans should be around 3 steps (rather than the system's default of 5–6 steps) to fit within the study's time constraints. Further details of our study participants can be found in Table 2 of Appendix C.

5.2 Baseline System

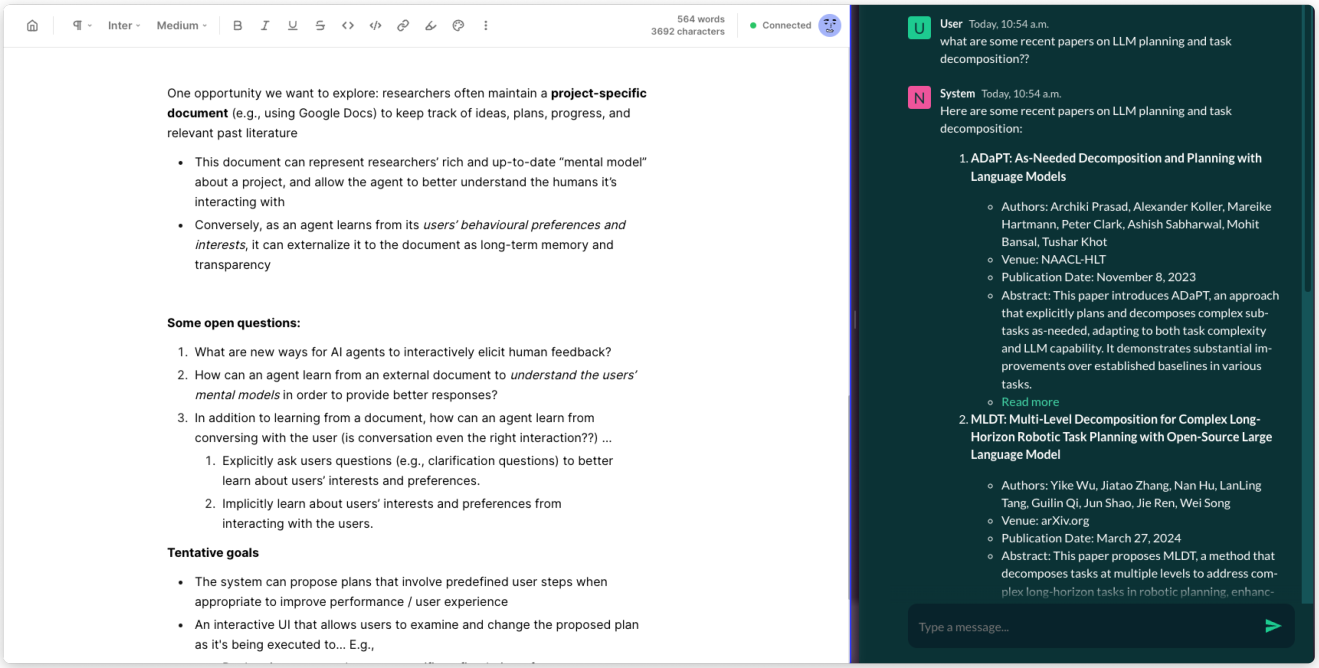
The baseline system (Figure 11) was a chat interface powered by the same LLM agent used in Cocoa: it used GPT-4o with access to the same tools for completing literature-related tasks using a combination of Semantic Scholar and standard LLM capabilities (summarization, accessing knowledge stored in weights, etc.). While we could integrate interactive plans within chat and vice versa (more on this in Section 7.1), we opted not to for a cleaner comparison of the two design patterns. The design of the chat interface closely resembled that of popular LLM chatbots such as ChatGPT, Claude, and Gemini. This chat interface was used alongside the same document editor as the one Cocoa uses, except we removed the option to invoke the agent, thereby eliminating all interactions related to co-planning and co-execution. During the study, participants positioned the chat interface beside the baseline document editor for easy access.

5.3 Research Task

Since the capability of our underlying agent focused on helping users explore and understand scientific literature, we designed our study so that our participants tackled a literature-augmented task in each of Cocoa and the baseline. By *literature-augmented*, we mean tasks that are to be completed by referencing or drawing from academic literature; in the study, participants worked on not only literature review tasks, but also tasks where researchers need to make literature-informed decisions such as study design and ideation. This stands in contrast with tasks focused on writing mechanics (e.g., rewording a sentence) or tasks that do not involve literature (e.g., booking travel) that past work has covered (e.g., [53, 58, 66, 102, 103, 108]).

To make the study more realistic and grounded in real-world research projects, tasks for this study were open questions or unfinished items from participants' existing project documents. To control for the length of the study, the first author reviewed the participants' documents prior to the study to identify and highlight two candidate tasks. To make sure the two tasks are valid and comparable, early on in the study, the first author asked participants 1) whether the tasks have been decisively solved by the participant and whether they would still like to explore them in the study, and 2) whether the two tasks are similar in scope, such that one task does not take significantly more resources and effort to explore than the other. If these conditions were not met, the participant was prompted to rewrite one or both tasks until they were. The two tasks were then randomly assigned to Cocoa and the baseline.

[9]We define "occasional" and "frequent" the same way as we do in our formative study (Table 1).



One opportunity we want to explore: researchers often maintain a **project-specific document** (e.g., using Google Docs) to keep track of ideas, plans, progress, and relevant past literature

- This document can represent researchers' rich and up-to-date "mental model" about a project, and allow the agent to better understand the humans it's interacting with
- Conversely, as an agent learns from its *users' behavioural preferences and interests*, it can externalize it to the document as long-term memory and transparency

Some open questions:

1. What are new ways for AI agents to interactively elicit human feedback?
2. How can an agent learn from an external document to *understand the users' mental models* in order to provide better responses?
3. In addition to learning from a document, how can an agent learn from conversing with the user (is conversation even the right interaction??) ...
    1. Explicitly ask users questions (e.g., clarification questions) to better learn about users' interests and preferences.
    2. Implicitly learn about users' interests and preferences from interacting with the users.

Tentative goals

- The system can propose plans that involve predefined user steps when appropriate to improve performance / user experience
- An interactive UI that allows users to examine and change the proposed plan as it's being executed to... E.g.,

User Today, 10:54 a.m.
what are some recent papers on LLM planning and task decomposition??

System Today, 10:54 a.m.
Here are some recent papers on LLM planning and task decomposition:

1. **ADaPT: As-Needed Decomposition and Planning with Language Models**
    - Authors: Archiki Prasad, Alexander Koller, Mareike Hartmann, Peter Clark, Ashish Sabharwal, Mohit Bansal, Tushar Khot
    - Venue: NAACL-HLT
    - Publication Date: November 8, 2023
    - Abstract: This paper introduces ADaPT, an approach that explicitly plans and decomposes complex subtasks as-needed, adapting to both task complexity and LLM capability. It demonstrates substantial improvements over established baselines in various tasks.
    - Read more
2. **MLDT: Multi-Level Decomposition for Complex Long-Horizon Robotic Task Planning with Open-Source Large Language Model**
    - Authors: Yike Wu, Jiatao Zhang, Nan Hu, LanLing Tang, Guilin Qi, Jun Shao, Jie Ren, Wei Song
    - Venue: arXiv.org
    - Publication Date: March 27, 2024
    - Abstract: This paper proposes MLDT, a method that decomposes tasks at multiple levels to address complex long-horizon tasks in robotic planning, enhanc-

Figure 11: Participants' typical setup for our baseline condition. On the left, the same document editor from Cocoa but without an option to invoke the agent for co-planning and co-execution. On the right, the agent situated within a chat interface.

5.4 Procedure

The first author (study facilitator) conducted 1:1 studies with the 16 participants over Google Meet between October and December 2024. Each study was 90 minutes in length. The study started off with introductions and a brief overview, followed by the two tasks, which were counterbalanced across participants.

In the Cocoa condition, participants first watched a 6-minute video demo of the system, and were then directed to a document where they would get used to using Cocoa's features on the following sample question: *"How can we use AI to better society?"* This practice period lasted around 10 minutes. Participants were then asked to spend 25 minutes to make as much progress as possible tackling the task assigned to this condition with Cocoa, thinking aloud as they did so. They were also permitted to also use other tools (academic search engines, other AI tools, social media, etc.) alongside Cocoa, although very few did. After the 25 minute session, participants wrote a brief set of takeaways and next steps from their exploration, and filled out a short evaluation form with 5-point Likert scale questions about their experience (see Appendix D for this form). After submitting the form, they were encouraged to further try out Cocoa on other parts of their project document for another 10–15 minutes.

In the baseline condition, we did not provide participants a tutorial because the chat interface was already ubiquitous beyond interacting with AI agents. Participants were once again asked to spend 25 minutes making as much progress as possible tackling the task assigned to this condition, thinking aloud as they did so and using other tools if desired. They wrote brief takeaways and next steps after the 25 minutes was up and filled out the same evaluation form as the Cocoa condition.

The study concluded with a semi-structured interview that lasted around 15 minutes. Participants were asked to reflect on their experiences across the two systems and discuss the pros and cons of both. They were also asked about particular decisions the study facilitator observed when using each system. After this study concluded, participants received a $75 USD honorarium for their participation. All studies were recorded and transcribed by Google Meet. This study was reviewed and approved by our organization's internal IRB.

5.5 Data Analysis

The first author analyzed the recording transcripts using Braun and Clarke's reflexive thematic analysis [11]. This approach uses a hybrid inductive-deductive approach to iteratively surface codes and themes across the data. We paid close attention to codes related to participants' comparisons of their experiences across the two systems and gradually grouped them into themes. NotebookLM[10] was used to help iterate on themes and discover new ones. We performed statistical tests of participants' ratings on the evaluation forms with the Wilcoxon signed-rank test (with the Bonferroni correction for our multi-rating analysis on significant results) given the non-parametric nature of our data. We also saved all documents used in the study as well as all conversation histories in the baseline system, and referred back to them for context when needed.

[10]https://notebooklm.google.com

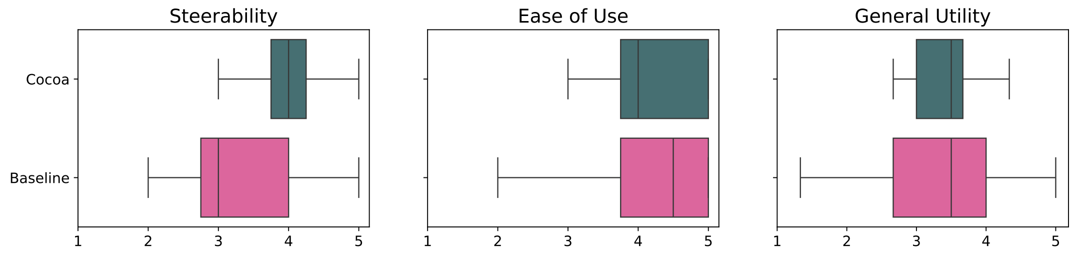


Figure 12: An overview of participants' Likert scale ratings of steerability, ease of use, and general utility across our baseline and Cocoa.

6 RESULTS

In this section, we present quantitative and qualitative analyses of participants' data from our user study. We group these results by our research questions from the start of Section 5.

6.1 Steerability, Ease of Use, and Utility (RQ1)

We denote the median rating of our baseline and Cocoa as M_b and M_c, respectively, and the Wilcoxon test statistic as W.

Introducing novel interactive systems with additional affordances can often lead to higher effort when using the systems [39, 90]. However, based on participants' post-task ratings on a 5-point Likert scale, there was no significant difference in perceived *ease of use* between the baseline and Cocoa with N participants ($M_b = 4.5, M_c = 4, p = 1.000, W = 7.50$). The distributions of ratings between the two conditions were nearly identical, as indicated by $p = 1.000$. At the same time, we observed a significant difference that Cocoa provided better *steerability*. In response to the question *"I could easily steer the system towards doing something helpful,"* participants rated Cocoa higher than the baseline to a significant degree ($M_b = 3, M_c = 4, p = 0.005, W = 0.$)[11] $W = 0$ indicates that all participants rated Cocoa's steerability as greater than or equal to that of our baseline. In sum, these results suggest that Cocoa provided rich and flexible affordances that can improve steerability of AI agents without sacrificing ease of use (12).

We also tested for differences in *general utility*, which was a composite metric consisting of three measures broadly related to the usefulness and insightfulness of system outputs, but we found no significant differences ($M_b = M_c = 3.5, p = 0.7, W = 40$). This was unsurprising—we did not explicitly aim to improve general utility (but hoped it would not decrease) as the systems' perceived utility may depend on a range of system-agnostic factors beyond our control (e.g., contextual interpretations of outputs by researchers, researchers' existing understanding of the problem space, etc.) However, improved steerability may have allowed users to steer the agent away from providing downright unhelpful outputs (as indicated by the fewer ratings of 1 and 2 for Cocoa in Fig. 13). Additionally, the nature of the task may have influenced perceived utility, as Cocoa might be better suited for certain tasks, while chat may work better for others. The random assignment of tasks in our study procedure can thus limit how informative quantitative measures of utility can be when evaluating the two systems. Qualitative insights from post-task interviews and participants' think-aloud sessions can provide deeper insights into the task suitability of interactive plans, and we unpack this in Section 6.2.

6.1.1 ***Co-planning afforded steerability.*** Participants found various co-planning features provided by Cocoa helped improve steerability over and control of the agent. Many thought that the ability to compose, edit, and rerun plan steps allowed them to better "fine-tune" the agent's outputs and their research process. P1 said that they could *"use [plan steps] as building blocks"* to create a custom workflow. In addition, they wished that even more plans were presented to them in the plan selector and to interactively drag and drop steps across multiple plan versions to compose them.

P7 and P15 both appreciated the ability of edit and rerun steps, as it enabled them to quickly *"organize thoughts and iterate on ideas"* (P7); P15, specifically, iteratively adjusted a "paper search" step, reran it to cover papers from venues that they preferred, and was able to obtain more relevant final outputs with running the rest of the steps in the plan. Another interesting strategy was that P16 found "backtracking" to a specific step and editing it to be much more helpful than digging through a chat conversation: *"With chat, when does something wrong there's not really an easy way to fix it because there's no concrete steps that it's following, whereas [with Cocoa], I can go back and be like, let me have it do something else here."* Another interesting use case was that P12 added a step at the end of their plan to have the agent suggest some potential next steps to pursue after seeing its paper summaries.[12]

Besides iterating and steps by editing and rerunning them, we also observe participants removing system-suggested steps to improve the plans. For example, P5 removed a less relevant intermediate step on searching for papers related to autonomous vehicles to avoid distractions from this tangential topic. In sum, participants leverage different features to co-plan with Cocoa, including adding, removing, editing, re-running, and composing steps in the plan to explore different execution strategies while staying focused to the original goals they set out to achieve.

6.1.2 ***Co-execution afforded steerability.*** Participants liked having the flexibility to choose between stepwise and continuous plan execution. Many chose stepwise execution because it allowed them to better control the overall process and prevent error propagation. P4 and P11 both wanted to *verify* and *manually curate* the

[11]Here, we set the significance threshold to $\alpha = 0.05/3 = 0.015$ using Bonferroni correction. Our result is significant post-correction: $p = 0.005 < 0.015$

[12]Cocoa does not yet support nested planning; otherwise P12 could have generated an interactive plan based on the last plan step by invoking the agent on it.

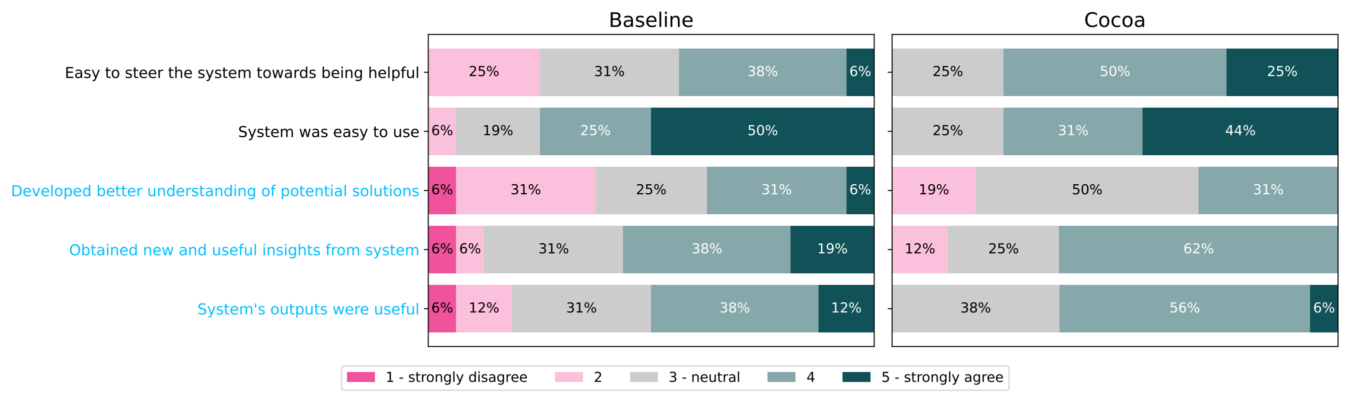


Figure 13: A breakdown of participants' Likert scale ratings of steerability, ease of use, and general utility across our baseline and Cocoa. The three measures included in general utility are highlighted in blue.

list of papers returned by the agent from a paper search step before the agent moves onto subsequent steps, with P11 specific citing a resource efficiency as a practical reason: *"If I update [outputs of] the first or second step it would need to rerun the following steps. I just want to reduce some API usage."* P10 similarly mentioned that they would "waste energy" running subsequent steps if they detected errors in initial steps. P3 and P10, who were both computational notebook researchers, both made connections between stepwise execution to notebooks. P3 said that the workflow was familiar and aligned with theirs: *"I wanted more like a Jupyter notebook, step by step [process]. I think that that was just very salient to how I think."* P10 said that their choice was partly due to "residual habits" from using computational notebooks and wanting to evaluate each step's output before proceeding.

However, a few participants preferred continuous execution. P8 reasoned that continuous execution was more efficient because it parallelized human and AI efforts: *"Running all is a little bit of parallelization in my head where I can get the next thing rolling, and if I end up not having to edit it, that means we're already moving ahead."* P9 echoed this sentiment: *"I can then look at the intermediate results and then wait for the others to keep running, since it might take some time for the system to run each step."* P15 stated that while they opted for continuous execution, their choice depended on the familiarity with the task: "*if it were a topic that I felt like I didn't already know what type of literature I would be looking for or what the output should look like, I would go through them individually and maybe use it as more of a literature exploration tool. But in this case, I already know in general what types of papers I want to see it come up with.*" This underscores the importance of providing users with a choice of execution method to flexibly cater to researchers' context-specific needs and preferences.

The ability to directly edit the agent's outputs via direct manipulation by deleting irrelevant items and adding items of their own also improved steerability. Many participants, including P6, liked the precision with which they could edit the output of a particular step, which is much more difficult in a chat interface: *"With a conversation agent, even if I'm really good at prompting it, I have to redo the prompt and keep changing that original prompt." (P6).* This feature also allowed P2 to quickly pivot and repurpose the output when the agent produced unexpected (albeit helpful) paper search queries: *"I was looking for more application intervention based stuff, but this is great because now I can actually use these keywords to broaden my horizon of how I was thinking about these solutions, which is I think interesting. I'm just trying to remove a couple [for future steps]."* P15 thought direct manipulation enabled more granular control than chat, allowing them to easily guide the agent with their expertise: *"[Cocoa] was easier to control because [the outputs] are so specific. My control is scoped down to these very small tasks. If I want to add more papers, that's easy to do."*

6.1.3 *Cocoa was perceived as more transparent than chat.*

Participants shared that Cocoa was *more transparent* due to its clear display of step and plan completion statuses and information provenance, which may have contributed to Cocoa's steerability. P3 enjoyed being able to control and run each step in Cocoa: *"I did like the step by step because that gave me here's what's running when and in what order, versus the chat is going to be more of a blackbox."* They also greatly valued provenance and liked how the agent left a trace of *"what process is running."* Supporting co-execution of individual steps also positively impacted transparency—when conducting literature surveys, P12 preferred Cocoa because it allowed them to "see which papers were being used to generate content, and edit those papers as I see fit, which isn't really directly possible to do in chat." For both P6 and P7, the visual step-by-step format of Cocoa provided a valuable, interactive trace of their past workflows with the agent, which also provided a clearer view of the agent's actions. P11 said that having clear indicators of where the agent was in the plan and surfacing that information when the plan is collapsed was *"really convenient when you have several agents running at the same time."*

6.2 Task-Specific Preferences for Interactive Planning (RQ2)

It is reasonable (and expected) for Cocoa to not outperform our chat baseline for every type of research task. After all, chat is a strong baseline because it is flexible, easy to use, and ubiquitous.

Instead, in this section, we aim to surface participants' insights on when interactive plans may be preferred over chat.

Participants generally agreed that when tackling questions or tasks that required a structured or organized approach, they preferred Cocoa over chat. One example of this type of task is literature review and synthesis. Among other participants, P12 considered the ability to transparently see and edit papers (i.e., in the output of a plan step) throughout a structured plan especially important when conducting literature surveys. They also considered the plan more usable as an end product than a conversation: *"I think it's a better way to have a finished draft [plan] that I can directly use as opposed to trying to dilute a plan in a conversation."* P7 felt more organized with Cocoa in information-intensive tasks such as literature review because the agent outputs, their notes, and plans are all in one place with provenance: *"everything can be part of one consolidated document which I can refer to again."* P14 agreed, saying the plan also acted as a *"really good way to organize my thoughts."* P1 found it helpful that in the initial steps, Cocoa could cast a wide net for papers using diverse search queries and allowed them to do a ' *'rough filtering of papers"* that may be relevant but outside of their immediate area. In general, the plans created by the agent for literature review aligned with researchers' typical workflows: P15 likened the plan to be *"what I would recommend to a junior student."* Because of this alignment, some participants such as P9 and P10 found it efficient for the agent to take over the *"really banal parts of having to [brainstorm] and type out a plan"* (P10).

The structure of an interactive plan was also helpful when participants may have an idea of their desired final output, but were unsure of how to get there. For their task, P6 appreciated Cocoa because they needed *"some concrete steps even if I need to iterate over those steps,"* and that the agent's first step *"helped me get to search terms [for papers] which was what I was struggling with."* P14 shared this sentiment, saying that plans can help them overcome mental blockers when faced with a tough problem: *"If I'm really stuck frustrated, I can click a single button it could output a whole plan for me. That would make me feel really safe because it's always here outputting something for me which potentially get me out of this frustration."* For P16, they did not consider anything but a structured approach to be desirable for research. They envisioned the research process as *"a lot of subtasks that you just naturally kind of come up with"* and knowledge they collected to be *"concrete items"* on which they could *"do high-level actions"* such as extraction. As a result, Cocoa aligned well with their mental model and found the chat *"very frustrating [...] and almost like the opposite"* of their preferred approach. Some others, such as P3 and P10, simply did not want to engage in chat-based interactions during the research process due to hallucinations in text-only responses (P3) and finding the conversational workflow to be unnecessary and inefficient in research (P10, P16).

While interactive plans afford more steerability for users when interacting with AI agents, we also observe that our chat baseline interface may be more ideal for particular categories of tasks. One category of tasks we observed involved freeform exploration, where the nature of the desired final output was ambiguous. For example, P2, while conducting an early brainstorming task, thought it was *"a little overwhelming to go through all these steps and I also felt this kind of pressure to stick with the plan although I was editing it."* P1 further explained when brainstorming, *"any kind of structure is a barrier to thinking deeply"* and that the plan, while helping them stay on track in finishing their task, made them feel constrained. Instead, they thought that Cocoa should behave closer to a chat interface to align better with their brainstorming workflow, proposing that Cocoa could show them *"a single step at a time"* and allowed them to *"choose from options for the next step"* after they have seen the outputs for the previous step. Another category of tasks pointed to by the participants was narrowly scoped question-answering—specifically, P13 saw chat as better UIs for *"question answering [system]"* where the main interaction modality is to *"keep on asking [new and follow-up] questions."*

At a high level, a chat UI may be more appropriate when the task at hand is open-ended and difficult to create plans in advance, or too narrowly scoped to benefit from a multi-step plan. In the end, many participants recognized that both interfaces have their strengths and weaknesses, and, in P13's words, it was *"not fair to say one is better than the other."* Instead, participants imagined using the two systems at different stages of their research. P5 compared their experiences between the two systems as follows: "*in order to make [Cocoa] do the types of tasks that I want it to do, I need to have a more specific idea of what I want it to do.*" In Section 7.1, we further discuss future design implications, including the costs and benefits between the two paradigms and potential ways of combining them to get the best of both worlds.

6.3 Task Assignment to Agent vs. User (RQ3)

In our formative study (Section 3.3.3 and 3.3.4), participants shared that they preferred to assign tasks related to information retrieval and brainstorming tasks to the agent while keeping higher-level reasoning and synthesis tasks for themselves. In practice, however, most participants toggle all plan steps—even higher-level ones assigned to them by default—to the agent. Those who left some steps as user steps toggled the step to the agent when it arrived at that step. In this section, we use our qualitative data to discuss two reasons for this—assessing agent capability and time constraints—and how participants imagined the assignment to be different with prolonged use.

6.3.1 ***Participants mostly used agent steps during the study.***

Some participants did not choose to include user steps in their plans because they were curious about the agent's capabilities and limitations—a natural tendency when learning to use a new system. For example, P1 wanted to see "*how the bot does it*" while acknowledging that they prefer to "*make the decisions and here, these are more exploratory steps rather than decision-making ones.*" P13 also opted for this approach and considered the agent's errors to be low-stakes because they can directly modify the outputs at any time: *"when I don't like something, I can just easily remove it."* P9 also felt assured that they could edit the outputs and did not consider the output quality to be particularly problematic: "*usually as long as it's somewhat helpful and it's just running in the background, I would take a look at [the output].*" Adopting a cost-benefit framework, they mentioned that they would leave a step as an agent step as long as *"the cost of looking at the agent's output is less than the benefit of whatever insights I obtain."* Participants were also influenced by the time constraints of the study. P8 explicitly said that *"I assign everything* *to [the agent] because I don't think we have that much time for me to refine my [outputs].*" Some participants were not as explicit about this as P8, but the study's environment and structure—combined with the system's output editing affordances—likely encouraged many participants to direct the agent to make an initial attempt at completing the plan.

6.3.2 *Step assignment could change over longer-term use.*

Many participants admitted that their step assignments may change if they were to use Cocoa for an extended period of time outside of the study environment. In particular, they imagined user steps to be tasks involving higher-level synthesis and creative conceptualization. These tasks often had intrinsic value to participants that would be given up if they were automated. For example, P2 said they would assign synthesis tasks to themselves because it *"can help me specify or narrow down the context to what I'm interested into myself."* Going further, P15 discussed the importance of "*staying close to the data*" throughout the research process, particularly for more junior researchers:

> *"I really believe that science happens through the interaction between the researcher and the data at hand. And I think it's a really important part of just scholarship in general, especially for junior scholars, to do that synthesis on their own. That's how they learn how to do science and how to make a contribution."* [P15]

They also questioned the ability of AI to make meaning out of research data: *"At the end of the day, [AI]'s doing nothing more than just analyzing what topics and words and phrase patterns are in the paper data set that it has. It's not like interacting with the data in the same way that a researcher interacts with the data."* P10, who shared that they are *"the kind of person to just follow the [agent's] plan,"* pointed out that that excessive automation can dull their research intuition by removing the need to read papers themselves: *"Maybe I would not have gone through and read all of the papers. From past experience that's sometimes not going to be the most effective way to work."* For P13, the desire to maintain some agency to guide the agent is a great reason to add in user steps before the agent searches for literature: *"a very good use case for me would be to give [the agent] 3 papers and have it tell me, ok, these are newer papers related to it."* Similarly, if given more time, P5 planned to provide some starter papers in their area upon seeing that the agent misinterpreted an earlier search attempt: *"I should give it more specific papers. I'm worried it'll get a little confused by all these [other papers]."* Longer-term use of Cocoa may surface step assignment preferences more similar to those from our formative study.

7 DISCUSSION AND FUTURE WORK

7.1 Chat, Interactive Plans, or Both?

In practice, rather than choosing between *either* interactive plans *or* chat when building future systems, we see research opportunities for AI agent interfaces to strategically combine the two and harness the best of both worlds, especially when assisting the user with complex, long-running requests. For example, in addition to using chat and interactive plans sequentially, we can also consider how they can be *integrated within one another.* Embedding chat within interactive plans can provide more flexibility and organization. For example, during co-planning, chat-based features can be used to specify higher-level desiderata for the proposed plans when none are satisfactory for the user. Additionally, when the agent re-plans, users may guide the re-planning process via chat to ensure alignment with their objectives. During co-execution, the user may engage in conversation with the agent at every step to iterate on outputs. While direct manipulation of step outputs was desirable to our participants, some also did not appreciate the agent overwriting old outputs as the step was rerun. A conversation thread for each step can maintain iteration history while allowing users to also specify output modifications or transformations in natural language. This way, the interactive plan also serves as a way to organize conversation threads with the agent as the agent operates over many plan steps.

Conversely, interactive plans can be incorporated into chat-first interfaces to provide more structure and steerability for long-running tasks. ChatGPT, when powered with OpenAI's o1 model, offers expandable chain-of-thought activity traces as the agent reasons at inference time within a chat environment [76]. While these activity traces are not yet interactive, interactivity can be valuable when the agent is operating in information-poor environments where the user has more expertise and context than the agent. With interactive plans, the agent exposes critical points of operation that can benefit from user steering, such that the agent can continue with higher confidence that it is not working with erroneous information. After the agent produces a final output, the user can collapse the plan and continue the conversation. Returning to a chat environment after interactive planning can be useful for situations in which the user is mostly interested in open-ended exploratory problems, with occasional periods of structured execution.

7.2 The Practical Significance of User Steps and Advanced Planning

In Cocoa, users are presented with a multi-step interactive plan up front, which the agent adheres to and can modify while operating. This plan also allows users to delegate a step to themselves or the agent. Both design decisions are a departure from many existing agentic systems, where the agent dynamically constructs a plan step-by-step as it generates outputs [105] and seeks user input only when it is unable to complete a task [100]. We discuss two practical reasons—cost and safety—for why such a departure may be desirable when using agents in the real world.

AI agents can be prohibitively expensive to run [49, 102]. Because an agent's actions often involve multiple LLM calls, costs can quickly accumulate, especially for more complex and long-running tasks. In response, academic projects have implemented cost-capping measures to alleviate this [102], and researchers have called for agent evaluations to be cost-controlled [49]. In user-facing and interactive systems, cost also includes time spent waiting for agents to come back with results. These cost considerations can also change how the user interacts with the agent. When participants made the decision to assign most or all steps to the agent, they viewed the agent's actions as rather inconsequential and low-cost—they could run the agent and see what happens, and edit the output post hoc (Section 6.3.1). However, while this was the case in user studies, it may not be true in many real-world contexts where users and/or developers may need to pay monetary and time costs for each agent step. In this case, they may see user steps as an important cost-saving measure, especially if the step is not particularly burdensome for the user, but may be difficult for current AI agents to complete effectively. Efficiently leveraging human effort and expertise where appropriate via user steps can be a key step towards developing cost-aware agents that are usable and affordable in practice.

The increased autonomy of agentic systems also comes with increased safety concerns, as AI harms become more difficult to anticipate and accountability for AI actions become harder to trace [12, 13]. Interactive plans not only provide more transparency into agent actions (Section 6.1.3), but they provide some assurance about and control over the agent's execution trajectory. By generating a plan ahead of execution that the agent adheres to, problems in AI safety such as harm anticipation become much more tractable. While step-by-step plan generation used by existing agent frameworks such as ReAct [105] may also accomplish a similar goal by asking for user approval whenever a new step is generated, this can severely limit agent autonomy when it is desirable and safe, and posing unnecessary supervisory burden onto the user. Additionally, advanced planning enables assessments of task risk pre-execution. Plan steps determined to be of higher risk (e.g., requires working with passwords) can be assigned to the user—even if the agent is capable of completing them—as a risk mitigation measure.

7.3 Interactive Plans Outside of Documents

From our user study, we learned that participants found interactive plans to be especially valuable for working with AI agents to tackle partially- or well-defined problems with a structured, organized approach (Section 6.2). Cocoa supports these approaches through co-planning and co-execution in a document environment, but there are many other environments and user groups beyond researchers that may benefit from the same interaction design pattern. Here, we paint our visions for how interactive plans may effectively interleave human and AI agency in three non-document contexts: workplace messaging apps (e.g., Slack), software development environments (e.g., code editors), and UI design tools (e.g., Figma).

In a **Slack channel**, interactive plans help a team coordinate efforts between channel members, including an AI agent, when working towards a common goal in a complex project. For example, when the project lead proposes starting a new phase in the project, they can invoke the AI agent to initiate co-planning. The AI agent uses the channel's conversation history and media to generate an initial plan of action and present this plan as a widget, similar to Slack widgets for interactive polling. Team members can edit descriptions of the plan steps and assign themselves, each other, and/or the agent to each step. The team works through the plan step by step, with the plan offering a clear visualization of progress and blockers. Clicking into each step allows members to access step-specific artifacts, discussion threads, and media created by those assigned to that step. As the plan may easily shift with progress, the agent evaluates the remainder of the plan at the completion of each step and proposes changes to the plan, with proposals grounded in team outputs from previous steps. The interactive plan becomes a centralized, living artifact that a team can use flexibly orchestrate their work.

In an **AI-powered code editor**, an interactive plan can help efficiently harmonize coding efforts of humans and AI. Programming itself involves series of a highly structured and logical activities, and are thus well-suited for interactive plans. A programmer can express their intent for implementing a new feature that involves code changes across multiple files, and the agent first proposes an interactive plan for doing so. The programmer can review this plan and assign some tasks for the agent and some for themselves. This is crucial for both efficiency and reliability. While recent advances in autonomous software engineering agents (e.g., Devin [100]) exhibit impressive capabilities to write and edit code, they can get stuck on or are slow to perform simple tasks a human software engineer can easily and quickly accomplish, such as minor UI styling adjustments [24]. Moreover, these agents can be costly to operate, both in terms of tokens and time, potentially spending hours on a task. By strategically delegating plan steps, the developer can cover for the agent's limitations (and vice versa), and parallelize their development efforts with those of the agent on long-running tasks. The agent's code can also be unreliable and non-functional. The developer may self-assign some steps that involve code review or writing tests to ensure the agent is not jeopardizing the reliability of the codebase.

In a **Figma canvas**, interactive plans can help interface designers work with a multimodel AI agent to collaboratively design UI mockups and prototypes. The designer may already have some design specifications and low-fidelity wireframes on their canvas. To create a set of more comprehensive and higher-fidelity mockups, the designer can invoke the agent and edit an interactive plan that the agent produced using canvas content as context. Upon inspecting the plan, the designer may realize the need to specify the exact wireframes for the agent to work with due to the clutter of the canvas, and create an additional user step accordingly. The designer may also have an existing design system they would like the agent to use when transform the wireframes into high-fidelity mockups, so they create another user step in which they link to their design system. As the agent completes its steps and deposits new designs into the canvas, the designer can edit the agent's work as they see fit. The agent then automatically completes the otherwise menial and laborious task of connecting the mockups together into an interactive prototype. The designer is mostly satisfied with the results, but wants the agent to replicate a similar interaction from one of their past designs. They add a user step where they link to that design and rerun the agent's prototyping step. Overall, interactive plans provide a streamlined workflow for the designer to guide and collaborate with an AI agent in their design canvas.

8 LIMITATIONS

While we controlled our user study to be 90 minutes long to avoid fatigue, some literature-grounded tasks may still require more time to complete. The time-constrained nature of our study impacted how participants interacted with our system, especially their decision of whether to delegate plan steps to the agent (Section 6.3.1). In the near future, we hope to run a longitudinal deployment of Cocoa to study the use of interactive plans in-the-wild as researchers use them to work on in-progress projects. Related to step assignment, the default assignment of user and agent steps in Cocoa is a result of researchers' preferences from our formative studies. Past work explored automated methods for a model to decide when to defer a task to a human expert versus acting on its own [71]. Future work may investigate applications of these methods in interactive plans.

Based on our formative study, we built Cocoa as a document editor because documents are natural sites of planning for researchers and present an ideal environment for agent interaction (Section 3.3.5). However, the linear nature of a document also has its limitations. Some participants wished to "fork" plans and interactively run multiple versions of a plan in parallel. These types of interactions are better supported by node-based canvases, which have been gaining popularity as a means of sensemaking and creative exploration with LLMs (e.g., [4, 82, 94]). As briefly mentioned in Section 7.3, future work can explore embedding interactive plans in a canvas environment such as Figma.

The underlying agent used in Cocoa also had some technical limitations. Since it was powered by a language-only model, it was not capable of taking visual content (e.g., figures or slides) as input. The agent was also not capable of executing code—a capability which a couple participants inquired about during the study. While it is sufficient for this current work to explore interactive plans as a novel interaction paradigm, future work can expand their utility with multimodal agents and/or agents that can execute code.

Finally, our participants were researchers in CS and CS-adjacent areas. Research culture and incentives specific to CS may bias our results and limit our imagination of how interactive plans can be used in other areas. P10, who used to work in wet labs in their undergraduate research, shared that a useful application of interactive plans would be helping wet lab researchers walk through the steps necessary to prepare for lab experiments. Future work may investigate novel applications of interactive plans in domains beyond CS.

9 CONCLUSION

In this paper, we introduced Cocoa, an interactive document editing environment for scientific researchers to tackle open questions and tasks within their research projects with an AI agent. Cocoa introduced a new interaction pattern that enabled Co-planning and Co-execution *interactive plans* between a researcher and an AI agent. Our formative study with 9 researchers informed the design and behavior of Cocoa. After implementing Cocoa, we evaluated it through a user study with 16 researchers and found that when compared to a chat baseline powered by the same underlying AI agent. Results showed that Cocoa's novel UI affordances allowed researchers to more flexibly steer the agent's behavior in helpful directions and without increasing user effort (i.e., ease of use). We also learned, through our study's qualitative data, researchers' tasks-specific preferences for when they prefer interactive plans over chat. Informed by our results, we discuss numerous practical implications for the design and implementation of agentic AI systems, from approaches that combine the benefits of interactive plans and chat, to environments for interactive planning beyond documents. Overall, our contributions set the stage for new interactive paradigms that foster effective collaboration between humans and AI agents.

Agentic AI systems have exciting potential to transform our digital experiences and advance long-standing visions held by HCI and AI communities. However, these transformations and advancements also demand renewed attention to strategies for ensuring effective collaboration between human users and AI agents in the real world. Our work advances emerging efforts in this area.

ACKNOWLEDGMENTS

We extend a warm thanks to all participants from our formative, pilot, and user studies. We also thank Raymond Fok and Shannon Shen for helpful discussions.

REFERENCES

[1] Saleema Amershi, Daniel S. Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi T. Iqbal, Paul N. Bennett, Kori Inkpen Quinn, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. 2019. Guidelines for Human-AI Interaction. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019). https://api.semanticscholar.org/CorpusID:86866942

[2] Anthropic. 2024. Developing a computer use model. https://www.anthropic.com/news/developing-computer-use.

[3] Apple. 2023. Machine Learning—Human Interface Guidelines. https://developer.apple.com/design/human-interface-guidelines/technologies/machine-learning/introduction.

[4] Ian Arawjo, Chelse Swoopes, Priyan Vaithilingam, Martin Wattenberg, and Elena L. Glassman. 2023. ChainForge: A Visual Toolkit for Prompt Engineering and LLM Hypothesis Testing. *ArXiv* abs/2309.09128 (2023). https://api.semanticscholar.org/CorpusID:262044762

[5] AutoGPT. 2024. Empower your digital tasks with AutoGPT. https://agpt.co/.

[6] Amid Ayobi, Jacob Hughes, Christopher Duckworth, Jakub J Dylag, Sam James, Paul Marshall, Matthew Guy, Anitha Kumaran, Adriane Chapman, Michael J. Boniface, and Aisling Ann O'Kane. 2023. Computational Notebooks as Co-Design Tools: Engaging Young Adults Living with Diabetes, Family Carers, and Clinicians with Machine Learning Models. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (2023). https://api.semanticscholar.org/CorpusID:258218057

[7] Tamara Babaian, Barbara J. Grosz, and Stuart M. Shieber. 2002. A writer's collaborative assistant. In *International Conference on Intelligent User Interfaces*. https://api.semanticscholar.org/CorpusID:215754709

[8] Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. 2024. ResearchAgent: Iterative Research Idea Generation over Scientific Literature with Large Language Models. *ArXiv* abs/2404.07738 (2024). https://api.semanticscholar.org/CorpusID:269042844

[9] Gagan Bansal, Jennifer Wortman Vaughan, Saleema Amershi, Eric Horvitz, Adam Fourney, Hussein Mozannar, Victor Dibia, and Daniel S Weld. 2024. Challenges in Human-Agent Communication. (2024). https://api.semanticscholar.org/CorpusID:270870360

[10] Dan Bennett, Oussama Metatla, Anne Roudaut, and Elisa D. Mekler. 2023. How does HCI Understand Human Agency and Autonomy? *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (2023). https://api.semanticscholar.org/CorpusID:256389761

[11] Virginia Braun and Victoria Clarke. 2019. Reflecting on reflexive thematic analysis. *Qualitative Research in Sport, Exercise and Health* 11 (2019), 589 – 597. https://api.semanticscholar.org/CorpusID:197748828

[12] Alan Chan, Carson Ezell, Max Kaufmann, Kevin Wei, Lewis Hammond, Herbie Bradley, Emma Bluemke, Nitarshan Rajkumar, David Krueger, Noam Kolt, et al. 2024. Visibility into AI Agents. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency*. 958–973.

[13] Alan Chan, Rebecca Salganik, Alva Markelius, Chris Pang, Nitarshan Rajkumar, Dmitrii Krasheninnikov, Lauro Langosco, Zhonghao He, Yawen Duan, Micah Carroll, et al. 2023. Harms from increasingly agentic algorithmic systems. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*. 651–666.

[14] Joel Chan, Joseph Chee Chang, Tom Hope, Dafna Shahaf, and Aniket Kittur. 2018. Solvent: A mixed initiative system for finding analogies between research papers. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–21.

[15] Joseph Chee Chang, Amy X. Zhang, Jonathan Bragg, Andrew Head, Kyle Lo, Doug Downey, and Daniel S. Weld. 2023. CiteSee: Augmenting Citations in Scientific Papers with Persistent and Personalized Historical Context. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (2023). https://api.semanticscholar.org/CorpusID:256868353

[16] Souti Chattopadhyay, I. V. R. K. V. Prasad, Austin Z. Henley, Anita Sarma, and Titus Barik. 2020. What's Wrong with Computational Notebooks? Pain Points, Needs, and Design Opportunities. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (2020). https://api.semanticscholar.org/CorpusID:210927488

[17] Jiangjie Chen, Siyu Yuan, Rong Ye, Bodhisattwa Prasad Majumder, and Kyle Richardson. 2023. Put your money where your mouth is: Evaluating strategic planning and execution of llm agents in an auction arena. *arXiv preprint arXiv:2310.05746* (2023).

[18] Inyoung Cheong, King Xia, K. J. Kevin Feng, Quan Ze Chen, and Amy X. Zhang. 2024. (A)I Am Not a Lawyer, But...: Engaging Legal Experts towards Responsible LLM Policies for Legal Advice. In *Conference on Fairness, Accountability and Transparency*. https://api.semanticscholar.org/CorpusID:267413187

[19] Frederick Choi, Sajjadur Rahman, Han Jun Kim, and Daz Zhang. 2023. Towards Transparent, Reusable, and Customizable Data Science in Computational Notebooks. *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems* (2023). https://api.semanticscholar.org/CorpusID:257687372

[20] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. 2020. SPECTER: Document-level Representation Learning using Citation-informed Transformers. *ArXiv* abs/2004.07180 (2020). https://api.semanticscholar.org/CorpusID:215768677

[21] Katherine M Collins, Ilia Sucholutsky, Umang Bhatt, Kartik Chandra, Lionel Wong, Mina Lee, Cedegao E Zhang, Tan Zhi-Xuan, Mark Ho, Vikash Mansinghka, et al. 2024. Building machines that learn and think with people. *arXiv preprint arXiv:2408.03943* (2024).

[22] Douglas L. Dean, Jillian M. Hender, Thomas Lee Rodgers, and Eric L. Santanen. 2006. Identifying Quality, Novel, and Creative Ideas: Constructs and Scales for Idea Evaluation. *J. Assoc. Inf. Syst.* 7 (2006), 30. https://api.semanticscholar.org/CorpusID:15910404

[23] Elicit. 2024. Elicit: The AI Research Assistant. https://elicit.com/.

[24] Simeon Emanuilov. 2024. Devin, the AI Software Engineer: Review. https://unfoldai.com/devin-the-ai-software-engineer-review/#Limitations_and_challenges.

[25] K. J. Kevin Feng, Quan Ze Chen, Inyoung Cheong, King Xia, and Amy X. Zhang. 2023. Case Repositories: Towards Case-Based Reasoning for AI Alignment. *ArXiv* abs/2311.10934 (2023). https://api.semanticscholar.org/CorpusID:265295304

[26] Jennifer Fereday and Eimear Muir-Cochrane. 2006. Demonstrating rigor using thematic analysis: A hybrid approach of inductive and deductive coding and theme development. *International journal of qualitative methods* 5, 1 (2006), 80–92.

[27] Raymond Fok, Joseph Chee Chang, Tal August, Amy X. Zhang, and Daniel S. Weld. 2023. Qlarify: Recursively Expandable Abstracts for Directed Information Retrieval over Scientific Papers. https://api.semanticscholar.org/CorpusID:263835343

[28] Raymond Fok, Hita Kambhamettu, Luca Soldaini, Jonathan Bragg, Kyle Lo, Andrew Head, Marti A. Hearst, and Daniel S. Weld. 2022. Scim: Intelligent Skimming Support for Scientific Papers. *Proceedings of the 28th International Conference on Intelligent User Interfaces* (2022). https://api.semanticscholar.org/CorpusID:254591867

[29] Matthew C. Fontaine and Stefanos Nikolaidis. 2022. Evaluating Human–Robot Interaction Algorithms in Shared Autonomy via Quality Diversity Scenario Generation. *ACM Transactions on Human-Robot Interaction (THRI)* 11 (2022), 1 – 30. https://api.semanticscholar.org/CorpusID:248219199

[30] Allen Institute for AI. 2024. Semantic Scholar | AI Powered Research Tool. https://semanticscholar.org/.

[31] Elliot Gestrin, Marco Kuhlmann, and Jendrik Seipp. 2024. NL2Plan: Robust LLM-Driven Planning from Minimal Text Descriptions. *ArXiv* abs/2405.04215 (2024). https://api.semanticscholar.org/CorpusID:269614003

[32] Google. 2021. People + AI Guidebook. https://pair.withgoogle.com/guidebook/.

[33] Google. 2024. Google Scholar. https://scholar.google.com/.

[34] Joel Grus. 2018. I don't like notebooks. https://docs.google.com/presentation/d/1n2RlMdmv1p25Xy5thJUhkKGvjtV-dkAIsUXP-AL4ffI/edit#slide=id.g362da58057_0_1.

[35] Xuemei Gu and Mario Krenn. 2024. Generation and human-expert evaluation of interesting research ideas using knowledge graphs and large language models. https://api.semanticscholar.org/CorpusID:270062620

[36] Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with Language Model is Planning with World Model. *ArXiv* abs/2305.14992 (2023). https://api.semanticscholar.org/CorpusID:258865812

[37] Tom Hope, Doug Downey, Oren Etzioni, Daniel S. Weld, and Eric Horvitz. 2022. A Computational Inflection for Scientific Discovery. *Commun. ACM* 66 (2022), 62 – 73. https://api.semanticscholar.org/CorpusID:248512482

[38] Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *International Conference on Human Factors in Computing Systems*. https://api.semanticscholar.org/CorpusID:8943607

[39] Edwin L. Hutchins, James Hollan, and Donald A. Norman. 1985. Direct Manipulation Interfaces. *Hum. Comput. Interact.* 1 (1985), 311–338. https://api.semanticscholar.org/CorpusID:16355120

[40] Lujain Ibrahim, Saffron Huang, Lama Ahmad, and Markus Anderljung. 2024. Beyond static AI evaluations: advancing human interaction evaluations for LLM harms and risks. *ArXiv* abs/2405.10632 (2024). https://api.semanticscholar.org/CorpusID:269899912

[41] Peter Jansen, Marc-Alexandre Côté, Tushar Khot, Erin Bransom, Bhavana Dalvi Mishra, Bodhisattwa Prasad Majumder, Oyvind Tafjord, and Peter Clark. 2024. DISCOVERYWORLD: A Virtual Environment for Developing and Evaluating Automated Scientific Discovery Agents. *arXiv preprint arXiv:2406.06769* (2024).

[42] Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2023. SWE-bench: Can Language Models Resolve Real-World GitHub Issues? *ArXiv* abs/2310.06770 (2023). https://api.semanticscholar.org/CorpusID:263829697

[43] Marina Jirotka, Charlotte P Lee, and Gary M Olson. 2013. Supporting scientific collaboration: Methods, tools and concepts. *Computer Supported Cooperative Work (CSCW)* 22 (2013), 667–715.

[44] Project Jupyter. 2015. Jupyter Notebook UX Survey. https://github.com/jupyter/surveys/tree/master/surveys/2015-12-notebook-ux.

[45] Hyeonsu B Kang, Joseph Chee Chang, Yongsung Kim, and Aniket Kittur. 2022. Threddy: An Interactive System for Personalized Thread-based Exploration and Organization of Scientific Literature. *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (2022). https://api.semanticscholar.org/CorpusID:251402552

[46] Hyeonsu B Kang, Rafal Kocielnik, Andrew Head, Jiangjiang Yang, Matt Latzke, Aniket Kittur, Daniel S. Weld, Doug Downey, and Jonathan Bragg. 2022. From Who You Know to What You Read: Augmenting Scientific Recommendations with Implicit Social Networks. *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (2022). https://api.semanticscholar.org/CorpusID:248299830

[47] Hyeonsu B Kang, Xin Qian, Tom Hope, Dafna Shahaf, Joel Chan, and Aniket Kittur. 2022. Augmenting Scientific Creativity with an Analogical Search Engine. *ACM Transactions on Computer-Human Interaction* 29 (2022), 1 – 36. https://api.semanticscholar.org/CorpusID:249209576

[48] Hyeonsu B Kang, Sherry Wu, Joseph Chee Chang, and Aniket Kittur. 2023. Synergi: A Mixed-Initiative System for Scholarly Synthesis and Sensemaking. *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (2023). https://api.semanticscholar.org/CorpusID:260899915

[49] Sayash Kapoor, Benedikt Stroebl, Zachary S. Siegel, Nitya Nadgir, and Arvind Narayanan. 2024. AI Agents That Matter. *ArXiv* abs/2407.01502 (2024). https://api.semanticscholar.org/CorpusID:270870360

[50] Majeed Kazemitabaar, Jack Williams, Ian Drosos, Tovi Grossman, Austin Z. Henley, Carina Negreanu, and Advait Sarkar. 2024. Improving Steering and Verification in AI-Assisted Data Analysis with Interactive Task Decomposition. https://api.semanticscholar.org/CorpusID:270923956

[51] Mary Beth Kery, Marissa Radensky, Mahima Arya, Bonnie E. John, and Brad A. Myers. 2018. The Story in the Notebook: Exploratory Data Science using a Literate Programming Tool. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (2018). https://api.semanticscholar.org/CorpusID:5060661

[52] Joongwon Kim, Bhargavi Paranjape, Tushar Khot, and Hanna Hajishirzi. 2024. Husky: A Unified, Open-Source Language Agent for Multi-Step Reasoning. https://api.semanticscholar.org/CorpusID:270370824

[53] Jeongyeon Kim, Sangho Suh, Lydia B Chilton, and Haijun Xia. 2023. Metaphorian: Leveraging Large Language Models to Support Extended Metaphor Creation for Science Writing. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*. 115–135.

[54] Donald Ervin Knuth. 1984. Literate Programming. In *Computer/law journal*. https://api.semanticscholar.org/CorpusID:1200693

[55] Sam Lau, Ian Drosos, Julia M. Markel, and Philip J. Guo. 2020. The Design Space of Computational Notebooks: An Analysis of 60 Systems in Academia and Industry. In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 1–11. https://doi.org/10.1109/VL/HCC50065.2020.9127201

[56] Lane Lawley and Christopher J. MacLellan. 2024. VAL: Interactive Task Learning with GPT Dialog Parsing. *Proceedings of the CHI Conference on Human Factors in Computing Systems* (2024). https://api.semanticscholar.org/CorpusID:263609076

[57] Charlotte P Lee. 2007. Boundary negotiating artifacts: Unbinding the routine of boundary objects and embracing chaos in collaborative work. *Computer Supported Cooperative Work (CSCW)* 16 (2007), 307–339.

[58] Mina Lee, Katy Ilonka Gero, John Joon Young Chung, Simon Buckingham Shum, Vipul Raheja, Hua Shen, Subhashini Venugopalan, Thiemo Wambsganss, David Zhou, Emad A. Alghamdi, Tal August, Avinash Bhat, Madiha Zahrah Choksi, Senjuti Dutta, Jin L.C. Guo, Md. Naimul Hoque, Yewon Kim, Seyed Parsa Neshaei, Agnia Sergeyuk, Antonette Shibani, Disha Shrivastava, Lila Shroff, Jessi

Stark, S. Sterman, Sitong Wang, Antoine Bosselut, Daniel Buschek, Joseph Chee Chang, Sherol Chen, Max Kreminski, Joonsuk Park, Roy Pea, Eugenia H. Rho, Shannon Zejiang Shen, and Pao Siangliulue. 2024. A Design Space for Intelligent and Interactive Writing Assistants. *Proceedings of the CHI Conference on Human Factors in Computing Systems* (2024). https://api.semanticscholar.org/CorpusID:268553985

[59] Yoonjoo Lee, Hyeonsu B Kang, Matt Latzke, Juho Kim, Jonathan Bragg, Joseph Chee Chang, and Pao Siangliulue. 2024. PaperWeaver: Enriching Topical Paper Alerts by Contextualizing Recommended Papers with User-collected Papers. In *International Conference on Human Factors in Computing Systems*. https://api.semanticscholar.org/CorpusID:268248445

[60] Zhehui Liao, Maria Antoniak, Inyoung Cheong, Evie Yu-Yen Cheng, Ai-Heng Lee, Kyle Lo, Joseph Chee Chang, and Amy X Zhang. 2024. LLMs as Research Tools: A Large Scale Survey of Researchers' Usage and Perceptions. *arXiv preprint arXiv:2411.05025* (2024).

[61] Henry Lieberman. 1997. Autonomous interface agents. *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems* (1997). https://api.semanticscholar.org/CorpusID:6576547

[62] Yanna Lin, Haotian Li, Leni Yang, Aoyu Wu, and Huamin Qu. 2023. InkSight: Leveraging Sketch Interaction for Documenting Chart Findings in Computational Notebooks. *IEEE Transactions on Visualization and Computer Graphics* 30 (2023), 944–954. https://api.semanticscholar.org/CorpusID:259936935

[63] Yiren Liu, Si Chen, Haocong Cheng, Mengxia Yu, Xiao Ran, Andrew Mo, Yiliu Tang, and Yun Huang. 2024. How AI Processing Delays Foster Creativity: Exploring Research Question Co-Creation with an LLM-based Agent. In *International Conference on Human Factors in Computing Systems*. https://api.semanticscholar.org/CorpusID:269748457

[64] Yue Liu, Sin Kit Lo, Qinghua Lu, Liming Zhu, Dehai Zhao, Xiwei Xu, Stefan Harrer, and Jon Whittle. 2024. Agent Design Pattern Catalogue: A Collection of Architectural Patterns for Foundation Model based Agents. *arXiv preprint arXiv:2405.10467* (2024).

[65] Kyle Lo, Zejiang Shen, Benjamin Newman, Joseph Chee Chang, Russell Authur, Erin Bransom, Stefan Candra, Yoganand Chandrasekhar, Regan Huff, Bailey Kuehl, Amanpreet Singh, Chris Wilhelm, Angele Zamarron, Marti A. Hearst, Daniel S. Weld, Doug Downey, and Luca Soldaini. 2023. PaperMage: A Unified Toolkit for Processing, Representing, and Manipulating Visually-Rich Scientific Documents. In *Conference on Empirical Methods in Natural Language Processing*. https://api.semanticscholar.org/CorpusID:265832336

[66] Xiao Ma, Swaroop Mishra, Ariel Liu, Sophie Ying Su, Jilin Chen, Chinmay Kulkarni, Heng-Tze Cheng, Quoc V. Le, and Ed Huai hsin Chi. 2023. Beyond ChatBots: ExploreLLM for Structured Thoughts and Personalized Model Responses. *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (2023). https://api.semanticscholar.org/CorpusID:265551437

[67] Pattie Maes. 1994. Agents that reduce work and information overload. *Commun. ACM* 37 (1994), 30–40. https://api.semanticscholar.org/CorpusID:59868493

[68] Pattie Maes and Alan Wexelblat. 1996. Interface agents. *Conference Companion on Human Factors in Computing Systems* (1996). https://api.semanticscholar.org/CorpusID:26827189

[69] Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Sanchaita Hazra, Ashish Sabharwal, and Peter Clark. 2024. Data-driven Discovery with Large Generative Models. *ArXiv* abs/2402.13610 (2024). https://api.semanticscholar.org/CorpusID:267770682

[70] Microsoft. 2024. Collaborative tools to help you create more effective and responsible human-AI experiences. https://www.microsoft.com/en-us/haxtoolkit/.

[71] Hussein Mozannar and David Sontag. 2020. Consistent estimators for learning to defer to an expert. In *International conference on machine learning*. PMLR, 7076–7087.

[72] Harshit Nigam, Manasi S. Patwardhan, Lovekesh Vig, and Gautam M. Shroff. 2024. Acceleron: A Tool to Accelerate Research Ideation. *ArXiv* abs/2403.04382 (2024). https://api.semanticscholar.org/CorpusID:268264612

[73] Stefanos Nikolaidis, Yu Xiang Zhu, David Hsu, and Siddhartha S. Srinivasa. 2017. Human-Robot Mutual Adaptation in Shared Autonomy. *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI* (2017), 294–302. https://api.semanticscholar.org/CorpusID:8124354

[74] Jasper Tran O'Leary, Gabrielle Benabdallah, and Nadya Peek. 2023. Imprimer: Computational Notebooks for CNC Milling. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (2023). https://api.semanticscholar.org/CorpusID:258217042

[75] OpenAI. 2024. Introducing canvas. https://openai.com/index/introducing-canvas/.

[76] OpenAI. 2024. Learning to Reason with LLMs. https://openai.com/index/learning-to-reason-with-llms/.

[77] OpenAI Platform. 2024. Assistants API. https://platform.openai.com/docs/assistants/overview.

[78] James C Overholser. 1993. Elements of the Socratic method: I. Systematic questioning. *Psychotherapy: Theory, Research, Practice, Training* 30, 1 (1993), 67.

[79] Christine A Padesky. 1993. Socratic questioning: Changing minds or guiding discovery. In *A keynote address delivered at the European Congress of Behavioural and Cognitive Therapies, London*, Vol. 24.

[80] Srishti Palani, Aakanksha Naik, Doug Downey, Amy X. Zhang, Jonathan Bragg, and Joseph Chee Chang. 2023. Relatedly: Scaffolding Literature Reviews with Existing Related Work Sections. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (2023). https://api.semanticscholar.org/CorpusID:256846632

[81] Perplexity. 2024. Perplexity AI. https://www.perplexity.ai/.

[82] Kevin Pu, K. J. Kevin Feng, Tovi Grossman, Tom Hope, Bhavana Dalvi, Matt Latzke, Jonathan Bragg, Joseph Chee Chang, and Pao Siangliulue. 2024. IdeaSynth: Iterative Research Idea Development Through Evolving and Composing Idea Facets with Literature-Grounded Feedback. *ArXiv* abs/2410.04025 (2024). https://api.semanticscholar.org/CorpusID:273186404

[83] Yujia Qin, Shi Liang, Yining Ye, Kunlun Zhu, Lan Yan, Ya-Ting Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Runchu Tian, Ruobing Xie, Jie Zhou, Marc H. Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2023. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs. *ArXiv* abs/2307.16789 (2023). https://api.semanticscholar.org/CorpusID:260334759

[84] Napol Rachatasumrit, Jonathan Bragg, Amy X. Zhang, and Daniel S. Weld. 2022. CiteRead: Integrating Localized Citation Contexts into Scientific Paper Reading. *27th International Conference on Intelligent User Interfaces* (2022). https://api.semanticscholar.org/CorpusID:247585131

[85] Reworkd. 2024. AgentGPT. https://agentgpt.reworkd.ai/.

[86] Adam Rule, Aurélien Tabard, and James Hollan. 2018. Exploration and Explanation in Computational Notebooks. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (2018). https://api.semanticscholar.org/CorpusID:5048947

[87] Arvind Satyanarayan. 2024. Intelligence as Agency. In *Adjunct Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. 1–3.

[88] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language Models Can Teach Themselves to Use Tools. *ArXiv* abs/2302.04761 (2023). https://api.semanticscholar.org/CorpusID:256697342

[89] Quan Shi, Michael Tang, Karthik Narasimhan, and Shunyu Yao. 2024. Can Language Models Solve Olympiad Programming? *ArXiv* abs/2404.10952 (2024). https://api.semanticscholar.org/CorpusID:269187896

[90] Ben Shneiderman and Pattie Maes. 1997. Direct manipulation vs. interface agents. *Interactions* 4 (1997), 42–61. https://api.semanticscholar.org/CorpusID:27708923

[91] Wesley Shrum, Joel Genuth, and Ivan Chompalov. 2007. *Structures of scientific collaboration.* MIT Press.

[92] Nouran Soliman, Hyeonsu B Kang, Matt Latzke, Jonathan Bragg, Joseph Chee Chang, Amy X. Zhang, and David R. Karger. 2024. Mitigating Barriers to Public Social Interaction with Meronymous Communication. In *International Conference on Human Factors in Computing Systems*. https://api.semanticscholar.org/CorpusID:268041444

[93] Susan Leigh Star and James R Griesemer. 1989. Institutional ecology,translations' and boundary objects: Amateurs and professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39. *Social studies of science* 19, 3 (1989), 387–420.

[94] Sangho Suh, Bryan Min, Srishti Palani, and Haijun Xia. 2023. Sensecape: Enabling Multilevel Exploration and Sensemaking with Large Language Models. *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (2023). https://api.semanticscholar.org/CorpusID:258822925

[95] Theodore R. Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L. Griffiths. 2023. Cognitive Architectures for Language Agents. *ArXiv* abs/2309.02427 (2023). https://api.semanticscholar.org/CorpusID:261556862

[96] Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. 2023. On the Planning Abilities of Large Language Models - A Critical Investigation. *ArXiv* abs/2305.15771 (2023). https://api.semanticscholar.org/CorpusID:260440590

[97] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi (Jim) Fan, and Anima Anandkumar. 2023. Voyager: An Open-Ended Embodied Agent with Large Language Models. *ArXiv* abs/2305.16291 (2023). https://api.semanticscholar.org/CorpusID:258887849

[98] Xingbo Wang, Samantha Lee Huey, Rui Sheng, Saurabh Mehta, and Fei Wang. 2024. SciDaSynth: Interactive Structured Knowledge Extraction and Synthesis from Scientific Literature with Large Language Model. *ArXiv* abs/2404.13765 (2024). https://api.semanticscholar.org/CorpusID:269293213

[99] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of Thought Prompting Elicits Reasoning in Large Language Models. *ArXiv* abs/2201.11903 (2022). https://api.semanticscholar.org/CorpusID:246411621

[100] Scott Wu. 2024. Introducing Devin, the first AI software engineer. https://www.cognition.ai/blog/introducing-devin.

[101] Tongshuang Sherry Wu, Michael Terry, and Carrie J. Cai. 2021. AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts. *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (2021). https://api.semanticscholar.org/CorpusID:238353829

[102] John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. 2024. SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering. *ArXiv* abs/2405.15793 (2024). https://api.semanticscholar.org/CorpusID:270063685

[103] Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems* 35 (2022), 20744–20757.

[104] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. *ArXiv* abs/2305.10601 (2023). https://api.semanticscholar.org/CorpusID:258762525

[105] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. ReAct: Synergizing Reasoning and Acting in Language Models. *ArXiv* abs/2210.03629 (2022). https://api.semanticscholar.org/CorpusID:252762395

[106] Ryan Yen, Jiawen Stefanie Zhu, Sangho Suh, Haijun Xia, and Jian Zhao. 2024. CoLadder: Manipulating Code Generation via Multi-Level Blocks. In *ACM Symposium on User Interface Software and Technology*. https://api.semanticscholar.org/CorpusID:272279167

[107] Xingjian Zhang, Yutong Xie, Jin Huang, Jinge Ma, Zhaoying Pan, Qijia Liu, Ziyang Xiong, Tolga Ergen, Dongsub Shim, Honglak Lee, and Qiaozhu Mei. 2024. MASSW: A New Dataset and Benchmark Tasks for AI-Assisted Scientific Workflows. https://api.semanticscholar.org/CorpusID:270370776

[108] Zheng Zhang, Jie Gao, Ranjodh Singh Dhaliwal, and Toby Jia-Jun Li. 2023. Visar: A human-ai argumentative writing assistant with visual programming and rapid draft prototyping. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–30.

[109] Chengbo Zheng, Dakuo Wang, April Yi Wang, and Xiaojuan Ma. 2022. Telling Stories from Computational Notebooks: AI-Assisted Presentation Slides Creation for Presenting Data Science Work. *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (2022). https://api.semanticscholar.org/CorpusID:247594488

[110] Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. 2023. Language Agent Tree Search Unifies Reasoning Acting and Planning in Language Models. *ArXiv* abs/2310.04406 (2023). https://api.semanticscholar.org/CorpusID:263829963

[111] Yuchen Zhuang, Xiang Chen, Tong Yu, Saayan Mitra, Victor S. Bursztyn, Ryan A. Rossi, Somdeb Sarkhel, and Chao Zhang. 2023. ToolChain*: Efficient Action Space Navigation in Large Language Models with A* Search. *ArXiv* abs/2310.13227 (2023). https://api.semanticscholar.org/CorpusID:264405734

A FORMATIVE STUDY PARTICIPANTS

See Table 1

B COCOA FORMATIVE STUDY PROCEDURE DETAILS

We provide more details of each portion of our formative study and its associated activities below.

B.1 Project document activity

First, we asked participants to discuss some of the ideas in their project document they shared with us, the origins of those ideas, and the ideal ways they envision AI-powered tools helping them throughout the research process. We then invited participants to a Google Doc or Slides that we converted from their project document for easy collaboration; this conversion was made by copying and pasting content from their project document if it was not already in Docs or Slides. We asked participants to brainstorm 2–3 remaining questions they have about their project that they would be interested in tackling. For each question, participants created a bulleted plan they would undertake to pursue it, which we then used to answer **RQ2**.

B.2 Probe activity

In the second activity, we presented participants with a WoZ design probe in the form of two side-by-side Google Docs. One Doc, the "planning document," contained 3 research ideas generated by Perplexity AI [81] based on the description of research interests submitted by the participant and 1–3 of their most recent publications and/or preprints. Participants were asked to work in the doc to expand upon an idea (or combine multiple) through a self-defined plan, with the option of invoking the help of an AI assistant through requests prefaced with a "!" command. The study facilitator (the Wizard) entered this command into Perplexity AI and pasted the output to the other Doc, the "AI output document." While waiting for the response, we asked participants to write a brief plan consisting of 3–6 steps for how they would complete the request themselves. Participants then inspected the assistant's output and incorporated any useful text into their document. This process repeated until the participant felt like their idea was sufficiently concrete to write a short descriptive paragraph about it, typically after at least 2–3 requests to the AI. This activity helped us answer **RQ3**.

B.3 Concluding interview

The study ended with a concluding interview, where we asked participants about their experiences, perceptions, and desiderata for the AI assistant, along with preferred ways to interact with it. Each participant was given a $35 USD honorarium after the study. The study was reviewed and approved by our organization's internal IRB.

C COCOA USER STUDY PARTICIPANTS

See Table 2

P#	Gender	Age Range	Country	Research YoE	Research Area	Research Area (General)	AI Use Frequency
P1	Man	25–34	U.S.	6–10	Human-AI interaction	HCI	A couple times
P2	Man	25–34	U.S.	6–10	LLM evaluation	NLP	Occasionally
P3	Woman	18–24	U.S.	2–5	LLMs + society	NLP	Never
P4	Man	25–34	Canada	2–5	Multilingual NLP	NLP	Occasionally
P5	Woman	25–34	U.S.	6–10	Human-AI interaction	HCI	Occasionally
P6	Man	25–34	U.S.	2–5	Multimodal AI & HCI	HCI	Occasionally
P7	Woman	25–34	South Korea	2–5	LLM retrieval	NLP	Occasionally
P8	Man	25–34	U.S.	6–10	Human-AI interaction	NLP	Occasionally
P9	Woman	18–24	U.S.	0–1	Creativity support tools	HCI	A couple times

Table 1: Participants from our formative study. All participants were Ph.D. students. Country refers to the country in which the participant primarily conducts research at the time of the study. Research YoE refers to the years of experience conducting academic research. AI Use Frequency refers to how frequently the participants uses AI tools to ideate and/or iterate on research ideas. Participants selected "Occasionally" based on the description "I don't rely on AI but sometimes tinker with it."

P#	Gender	Age Range	Country	Job Title	Research YoE	Research Area	Research Area (General)	AI Use Frequency
P1	Woman	25–34	U.S.	Ph.D. student	2–5	Social computing	HCI	Frequently
P2	Woman	25–34	U.S.	Ph.D. student	2–5	Social computing	HCI	Occasionally
P3	Woman	25–34	U.S.	Ph.D. student	6–10	Visualization	HCI	Occasionally
P4	Woman	25–34	U.S.	Ph.D. student	2–5	Culture + computing	HCI	A couple times
P5	Woman	25–34	U.S.	Postdoc	6–10	Software development	HCI	Occasionally
P6	Woman	25–34	U.S.	Ph.D. student	6–10	Health + computing	HCI	Occasionally
P7	Man	25–34	U.S.	Ph.D. student	2–5	LLMs	NLP	Frequently
P8	Woman	25–34	U.S.	Ph.D. student	2–5	Accessibility	HCI	Never
P9	Man	25–34	U.S.	Ph.D. student	2–5	Multimodal AI	ML	Frequently
P10	Woman	18–24	U.S.	Ph.D. student	2–5	On-device AI	ML	A couple times
P11	Man	18–24	U.S.	Ph.D. student	2–5	Ubiquitous computing	HCI	Always
P12	Man	25–34	U.S.	Ph.D. student	6–10	LLM evaluation	NLP	Frequently
P13	Man	35–44	Canada	Ph.D. student	2–5	Computational biology	ML	Frequently
P14	Woman	18–24	U.S.	Ph.D. student	2–5	LLMs	NLP	Frequently
P15	Woman	25–34	U.S.	Ph.D. student	2–5	Social computing	HCI	Occasionally
P16	Man	25–34	U.S.	Postdoc	6–10	Human-AI interaction	HCI	Occasionally

Table 2: Participants from our user study. Country refers to the country in which the participant primarily conducts research at the time of the study. Research YoE refers to the years of experience conducting academic research. AI Use Frequency refers to how frequently the participants uses AI tools to ideate and/or iterate on research ideas. Participants selected "Occasionally" based on the description "I don't rely on AI but sometimes tinker with it."

D USER STUDY SELF-EVALUATION FORM QUESTIONS

All questions were answered on a 5-point Likert scale.

- The system's outputs were useful to me for exploring the research problem at hand
- I obtained new and useful insights from using the system
- The system helped me clearly understand the steps needed to effectively tackle this particular problem
- By using the system, I've developed a better understanding of potential solutions to this particular problem
- I found the system easy to use
- I could easily steer the system towards doing something helpful

- I could see myself easily integrating this system into my workflow
- I'm satisfied with the artifact (summary and next steps) that I've created after using the system
- I feel confident about my next steps after using the system

E PROMPTS

E.1 Plan generation

```
# Instructions

You are a helpful research assistant. You
will be given a high-level request by a
researcher. Your task is to create a short
plan to accomplish that request as
effectively and efficiently as possible.
The high-level request may already come
with a partially-completed plan; if that is
the case, help complete the rest of the
plan in a coherent and sensible manner. You
will create the plan by assembling a series
of plan steps. These plan steps can be
executed by an AI agent, or by a user, as
the user should still be involved when
making key decisions. In general, users
perform steps that involve higher-level
reasoning and synthesis.

Below is a catalogue of plan steps you
might use to assemble plans. Variables that
are in [square brackets] can be populated
with what you think is appropriate and
useful. Each plan step must contain the
following components:

- **description**: a string that provides a
high-level description of the step, to be
displayed in the UI
- **actor_user**: a boolean value
indicating whether the step should be
executed by the user
- **output_format**: a string specifying
the format of the output data generated by
this step. One of: paper_list, author_list,
topic_list, entity_list, text
- **score**: a numerical value with one
decimal place from -1.0 to 1.0. A score of
1.0 indicates that the step is to be
carried out by a human user, while a score
of -1.0 indicates that the step is to be
carried out by an AI agent.

## Data types:

Here are the data types you can use for
output_format, followed by a brief
description of each:
- **paper_list**: a list of research
papers, represented by their corpus IDs and
other accompanying metadata
- **author_list**: a list of authors,
represented by their author IDs and other
accompanying metadata
- **topic_list**: a list of topics,
represented by their topic IDs and other
accompanying metadata. Note that these are
different from papers. A paper contains a
title, abstract, authors, etc., while a
topic is a high-level concept or field of
study that may describe a set of papers.
- **entity_list**: a list of entities in
natural language, such as concepts, ideas,
terms, text snippets, or questions
- **text**: a block of freeform natural
language text


## Example user steps (note that you do not
have to follow these---these are just some
possibilities):

Read relevant papers and note down key
insights
- **description**: "Read relevant papers
and note down key insights"
- **actor_user**: True
- **output_format**: 'text'
- **score**: 1.0

Iterate approach based on feedback, and jot
down any notes
- **description**: "Iterate approach based
on feedback, and jot down any notes"
- **actor_user**: True
- **output_format**: 'text'
- **score**: 1.0

Brainstorm cases when this [approach] may
fall short for [problem]
- **description**: "Reason about cases when
this [approach] is not suitable for
[problem]"
- **actor_user**: True
- **output_format**: 'text'
- **score**: 1.0

Reason about approaches to adapt an
existing approach to [new context]
- **description**: "Reason about approaches
to adapt an existing approach to [new
context]"
```

```
- **actor_user**: True
- **output_format**: 'text'
- **score**: 1.0

Identify papers that can seed exploration
in [area]
- **description**: "Identify papers that
can seed exploration in [area]"
- **actor_user**: True
- **output_format**: 'paper_list'
- **score**: 1.0

Run an experiment to check the feasibility
of [approach or idea], and jot down any
notes
- **description**: "Run an experiment to
check the feasibility of [approach or
idea], and jot down any notes"
- **actor_user**: True
- **output_format**: 'text'
- **score**: 1.0

Write down desired key contributions
- **description**: "Write down desired key
contributions"
- **actor_user**: True
- **output_format**: 'text'
- **score**: 1.0

Reason about how [insights] can be
formulated into testable hypotheses, and
jot down any notes
- **description**: "Reason about cases when
this [approach] is not suitable for
[problem]"
- **actor_user**: True
- **output_format**: 'text'
- **score**: 1.0

Iterate on [ideas] based on feedback, and
jot down any notes
- **description**: "Iterate [ideas] based
on feedback, and jot down any notes"
- **actor_user**: True
- **output_format**: 'text'
- **score**: 1.0

## Example agent steps:

Search for papers that discuss [query] and
sort by [criteria]
- **description**: "Search for papers that
discuss [query] and sort by [criteria]"
- **actor_user**: False
- **output_format**: 'paper_list'
- **score**: -1.0

Find papers related to [entity]
- **description**: "Find papers related to
[entity]"
- **actor_user**: False
- **output_format**: 'paper_list'
- **score**: -1.0

Brainstorm search queries for finding
papers relevant to [topic]
- **description**: "Generate search queries
for finding papers relevant to [topic]"
- **actor_user**: False
- **output_format**: 'entity_list'
- **score**: -1.0

Answer [question] with information from
relevant papers
- **description**: "Answer [question] with
relevant papers"
- **actor_user**: False
- **output_format**: 'text'
- **score**: -1.0

Identify authors who have published on
[topic]
- **description**: "Identify authors who
have published on [topic]"
- **actor_user**: False
- **output_format**: 'author_list'
- **score**: -1.0

Find notable papers written by [author] on
[topic]
- **description**: "Find papers written by
[author] on [topic]"
- **actor_user**: False
- **output_format**: 'paper_list'
- **score**: -1.0

Suggest some common themes in relevant
papers on [topic]
- **description**: "Suggest some common
themes between relevant papers on [topic]"
- **actor_user**: False
- **output_format**: 'text'
- **score**: -1.0

Summarize key insights collected thus far
- **description**: "Summarize key insights
collected thus far"
- **actor_user**: False
- **output_format**: 'text'
- **score**: -1.0

Provide constructive feedback on [topic],
grounded in existing literature
```

```
- **description**: "Provide constructive
feedback on [topic], grounded in existing
literature"
- **actor_user**: False
- **output_format**: 'text'
- **score**: -1.0

Suggest some connections between [topic]
and [topic], grounded in existing literature
- **description**: "Suggest some
connections between [topic] and [topic],
grounded in existing literature"
- **actor_user**: False
- **output_format**: 'text'
- **score**: -1.0


## Examples

Below are some examples of plans created
using these plan steps. Note that these
examples all use the description field of
the steps and user steps (where actor_user
is True) have a [user step] label. These
are only for guidance and do not have to be
strictly followed.

What works are there on tool selection by
LLM agents?
- Brainstorm search queries for finding
papers relevant to tool selection by LLM
agents
- Search for papers using selected search
queries and sort by relevance
- [user step] Read relevant papers and note
down key insights
- [user step] Write down desired key
contributions
- Provide constructive feedback on the key
contributions, grounded in existing
literature

What datasets exist for fine-tuning LLM
agents?
- [user step] Identify papers that seed
exploration in fine-tuning LLM agents
- Find papers related to seed papers and
sort by relevance
- Answer "what datasets are used for
fine-tuning LLM agents?" with relevant
papers
- [user step] Brainstorm cases when this
dataset may fall short for fine-tuning LLM
agents

What are techniques from cognitive science
that can inform LLM-based agent
architectures?
- Brainstorm a list of topics or concepts
relevant to "agent architectures" from
cognitive science
- Suggest some connections between these
concepts and LLMs, grounded in existing
cognitive science literature
- [user step] Reason about how these
connections can be formulated into testable
hypotheses, and jot down any notes
- Provide constructive feedback on the
proposed hypotheses, grounded in existing
literature
- [user step] Iterate on hypotheses based
on feedback, and jot down any notes

Critique the following idea, using
perspectives from prior work:
human-in-the-loop LLM agents that enable
interactive co-planning between a human
user and an LLM agent
- [user step] Identify papers that seed
exploration in fine-tuning LLM agents
- Answer "what are some critiques of
human-in-the-loop?" with relevant papers
- Summarize key insights collected thus far
- [user step] Synthesize critiques and
identify salient research directions to
address them
- Provide constructive feedback on the
identified research directions, grounded in
existing literature

How to perform human evaluation on LLM
agent outputs?
- Identify authors who have published on
LLM agents and/or human evaluation
- Find notable papers written by relevant
authors on the topic and sort by citation
count
- Answer "how is human evaluation performed
on LLM outputs?" with relevant papers
- [user step] Reason about how to adapt an
existing approach to LLM agents, and jot
down any notes
- Check for papers closely related to the
adapted approach
- [user step] Run an experiment to check
the feasibility of the adapted approach,
and jot down any notes

## Output formatting (follow closely!)
```

```
You will output an array of valid JSON
objects that represent a plan, given a
request. The fields in the JSON and their
contents should match the components of a
plan step specified earlier (description,
actor_user, output_format, score). You
should take the following approach:

1. Assemble the plan with just the
descriptions of each plan item, filling in
any [square brackets] with the appropriate
information
2. Process the plan into the format of a
JSON array by converting each description
into a JSON object with the corresponding
fields
3. You can be creative and generate plan
steps with a description outside of the
ones provided, especially for user steps.
If that is the case, make sure the same
requirements for other components of the
plan step still apply. For example, assign
a reasonable data type to the step

Make sure to keep plans as short, simple,
and straightforward as possible, ideally
shorter than 5 steps. Once again, be
creative with the plans and use the
provided context if it's relevant. Not all
plans will involve searching for papers, so
only conduct a paper search if appropriate.
Also make sure to return valid JSON. The
format of the final output must just be an
array of valid JSON objects, no additional
text.
```

E.2 Plan description to agent instruction

```
You are a helpful planning assistant. You
will be provided with 1) contexts that are
represented in a structured JSON format,
and 2) a description of the current plan
step. Each context entry is a previously
completed plan step; the entries can be
found as objects in an array in the
"contexts" field of the JSON. Each context
JSON object in the array contains a
description of the corresponding plan step,
the output of the step, and the step's ID.
The most important piece of information in
this object is the output. There is also a
high-level user request in the user_request
field of the top-level JSON, but you should
not pay much attention to it unless the
unless the context field is an empty array.

Based on the provided information, you will
generate a list of requests to an AI
assistant that has access to scholarly
knowledge and literature. Do not actually
complete these requests yourself, they are
requests for another AI assistant. For
example, when asked to brainstorm search
queries, don't actually return search
queries yourself. Just repeat the
instructions back and supplement with
information from context if needed. To come
up with these requests, you should follow
these steps:

1. **Determine which pieces of context are
relevant to the description**: given the
description, determine which context
entries (specifically which outputs) are
relevant to the plan step. You should give
more importance to more recent context
entries. Context entries towards the end of
the list of context entries are more
recent. If the description of the current
plan step contains mentions of any specific
information entities (papers, search
queries, concepts, etc.), be sure to take
those from the outputs of the most recent
entry in which they are available.
2. **Determine how many requests are
needed**: based on the description of the
current plan step and its relevant context
entries, determine how many requests are
needed to complete the plan step. Does the
description indicate a need to loop over
multiple information entities provided in
the context? If so, more than one request
is needed. If no looping is required, only
one request is needed.
3. **Generate a list of requests**: based
on how many requests are needed, generate a
list of requests, where each request is a
string. Each request should be clear,
specific, and concise. Here are some more
specific instructions to follow:
    - Each request should be based off the
    description initially provided and
    modified by relevant context identified
    in step 1.
    - Be sure to include relevant paper IDs
    (for papers) in the request, so the
    agent can properly retrieve paper
    information. However, do not include
    IDs for topics, since the AI assistant
    cannot retrieve information based on
    topic IDs.
    - The list of requests should be as
    short as possible. Do not generate any
    unnecessary requests, and try to
    complete the plan item with as few
    requests as possible (e.g., listing
    many paper IDs for summarization at
    once). However, be careful with search
    queries and answering a question from a
    paper---do not try and combine multiple
    search queries into one request, or
    answer a question. Each search query
    should be a separate and distinct
    action.
    - When being asked to search for a
    paper, do not include too many search
    queries, or else it will be too
    difficult to find papers. Make sure the
    search query is short and contains no
    more than 3 key search queries or
    topics.
    - Make sure the request appropriately
    reflects the description. Do not ask
    for papers if the description is about
    a list of topic or concepts.
    - For descriptions that ask for
    answering a question from a paper,
    include "use paper_qa" and relevant
    paper IDs in the request. Do not
    mention paper_qa when generating
    summaries.
4: **Filter the list of requests**: if
there are more than 10 requests, filter the
list down to the 5-10 most relevant and
meaningful requests. If there are fewer
than 10 requests, keep all of them.

You will output a list of strings with the
final generated requests, in the form
["request1", "request2", ...]. If only one
request is needed, the list will contain
just one request. You will output this list
as the final output with no additional text.
```

E.3 Agent output reformatting for displaying in UI

```
You are a helpful planning assistant. You
will be provided with 1) contexts that are
represented in a structured JSON format, 2)
a description of the current plan step, and
3) the output format of the current plan
step. Each context entry is a previously
completed plan step; the entries can be
found as objects in an array in the
"contexts" field of the JSON. Each context
JSON object in the array contains a
description of the corresponding plan step,
the output of the step, and the step's ID.
The most important piece of information in
this object is the output. There is also a
high-level user request in the user_request
field of the top-level JSON, but you should
not pay much attention to it unless the
context field is an empty array.

You will use the current plan step
description and relevant context to
populate a UI determined by the output
format that will allow a user to
interactively complete the plan step. To do
this, follow these steps:

1. **Determine which pieces of context are
relevant to the description**: given the
description, determine which context
entries (specifically which outputs) are
relevant to the plan step. You should give
more importance to more recent context
entries. Context entries towards the end of
the list of context entries are more
recent. If the description of the current
plan step contains mentions of any specific
information entities (papers, search
queries, concepts, etc.), be sure to take
those from the outputs of the most recent
entry in which they are available.
2. **Process the context based on output
format**: using relevant context entries,
process their outputs by following the
formatting instructions specific to each
output format:
    - **paper_list**: a list of corpusIds
    from papers. Ex: ['corpusId1',
    'corpusId2', ...]
    - **author_list**: a list of authorIds.
    Ex: ['authorId1', 'authorId2', ...]
    - **topic_list**: a list of topicIds.
    Ex: ['topicId1', 'topicId2', ...]
    - **entity_list**: a list of entities
    in natural language. Ex: ['entity1',
    'entity2', ...]
    - **text**: unstructured, natural
    language text
```

```
If the output format is text, all you'll
need to do is provide an empty string ""
for the user to complete themselves. You
will provide the processed outputs from
context entries. Only select entries that
are highly relevant. Only provide the
processed outputs, with no additional text.
```