Talk to the Hand: an LLM-powered Chatbot with Visual Pointer as Proactive Companion for On-Screen Tasks

Citation: Thanawit Prasongpongchai, Pat Pataranutaporn, Monchai Lertsutthiwong, and Pattie Maes. 2025. Talk to the Hand: an LLM-powered Chatbot with Visual Pointer as Proactive Companion for On-Screen Tasks. In Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25). Association for Computing Machinery, New York, NY, USA, Article 637, 1–16.

As Published: https://doi.org/10.1145/3706598.3715579

Publisher: ACM|CHI Conference on Human Factors in Computing Systems

Persistent URL: https://hdl.handle.net/1721.1/164288

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

Talk to the Hand: an LLM-powered Chatbot with Visual Pointer as Proactive Companion for On-Screen Tasks

Thanawit Prasongpongchai
Beacon Interface
KASIKORN Business-Technology Group
Nonthaburi, Thailand
thanawit.p@kbtg.tech

Pat Pataranutaporn
MIT Media Lab
Massachusetts Institute of Technology
Cambridge, Massachusetts, USA
patpat@media.mit.edu

Monchai Lertsutthiwong
KASIKORN Labs
KASIKORN Business-Technology Group
Nonthaburi, Thailand
monchai.le@kbtg.tech

Pattie Maes
MIT Media Lab
Massachusetts Institute of Technology
Cambridge, Massachusetts, USA
pattie@media.mit.edu

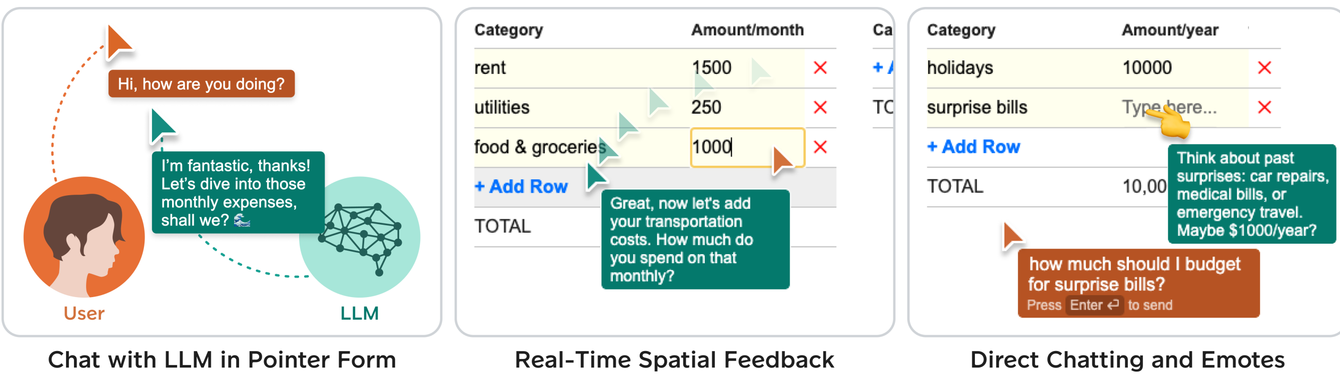


Figure 1: Example interactions of Pointer Assistant, an interaction technique for LLM-based human-AI interaction that features displaying the AI assistant as an animated extra mouse pointer (in green) alongside the user's pointer (in orange).

Abstract

This paper presents Pointer Assistant, a novel human-AI interaction technique for on-screen tasks. The design features a chatbot displayed as an extra mouse pointer, alongside the user's, which proactively gives feedback on user actions while directing them to relevant areas on the screen and responding to the user's direct chat messages. The effectiveness of the design's key characteristics, pointer form and proactivity, was investigated in a study involving 220 participants in a financial budget planning task. Results demonstrated that the pointer design and interaction reduced task load while improving satisfaction with the experience, and increased the number of budget categories ideated during the task compared to the traditional passive chat log design. Participants viewed Pointer Assistant as a fun, innovative, and helpful visual guide while noting that its assertiveness can be improved. Future developments could offer even further enhancements to the user experience of human-AI collaboration and task outcomes.

ACM ISBN 979-8-4007-1394-1/25/04
https://doi.org/10.1145/3706598.3715579

CCS Concepts

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

Keywords

Human-AI Interaction Technique, Large Language Models, Human-AI Collaboration, Pointing Devices, Real-Time Feedback

ACM Reference Format:
Thanawit Prasongpongchai, Pat Pataranutaporn, Monchai Lertsutthiwong, and Pattie Maes. 2025. Talk to the Hand: an LLM-powered Chatbot with Visual Pointer as Proactive Companion for On-Screen Tasks. In *CHI Conference on Human Factors in Computing Systems (CHI '25), April 26–May 01, 2025, Yokohama, Japan.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3706598.3715579

1 Introduction

The integration of Artificial Intelligence (AI) into everyday digital experiences has dramatically reshaped how users interact with technology. As AI assistants become increasingly prevalent across various domains, understanding the nuances of human-AI interaction is crucial for designing effective and user-friendly systems.

With the recent meteoric rise of Large Language Models (LLMs), traditional chat log–based interfaces—ones that feature a vertical chat log in a dedicated panel on the graphical user interface—have become the dominant design choice for interaction with chatbots. This design, while simple and easy to understand, often requires users to switch their attention between the task at hand and a separate chatbot window, potentially increasing cognitive load. Also, while chat log–based interactions can be used as a user-friendly interaction technique for short data collection tasks where the chatbot acts as an interviewer asking for information from the user, they can become tedious and time-consuming in more complex tasks. This challenge is particularly relevant in spatial tasks where users must navigate complex interfaces while considering multiple factors simultaneously.

Moreover, the popularization of LLMs has also sparked discussions about the role of AI in human-AI interaction. Popular implementations of LLMs have posited AIs as an assistant that obeys user input, even going so far as to perform autonomous task completion for the user (such as taking over the user's mouse pointer to complete a task[22]). In contrast, some researchers have argued for a fundamentally different direction that AI chatbots should be designed to be more proactive, inquisitive, or even critical of user's input [5, 6, 26], in order to ultimately enhance, rather than replace, human agency in human-computer interactions. This poses an opportunity where we might reconsider the role of an LLM chatbot by shifting it from an assistant we talk directly to in order to obtain or give information, toward being a collaborator that works simultaneously with us and gives us real-time feedback on a task at hand.

Motivated by these aspects, we draw inspiration from research on pointing devices and cursor interactions. The computer mouse, introduced by Engelbart in 1964, has become the de facto standard for cursor control [14]. However, researchers have been continually exploring alternative pointing devices and interaction techniques to improve efficiency, ergonomics, and accessibility. For instance, Cao et al. [3] demonstrated the potential of combining finger and hand movements for improved pointing performance, while Blanch and Ortega [2] introduced the rake cursor technique, which combines mouse input with gaze tracking to significantly outperform traditional mouse-only pointing. On the other hand, the original idea of the mouse cursor has also grown in consumer products over the years as can be seen in the development of real-time online collaboration software such as Miro [20] and Figma [11]. These applications feature the ability for multiple users to edit elements on the same "board" at the same time with other users being represented as cursors on the screen, shifting away from the pointing cursor being only a single-user experience.

Building on these advancements, our research explores how different configurations of chatbot behavior and visual cues can influence user experience, task performance, and decision-making. In this paper, we introduce Pointer Assistant, a novel interaction technique that integrates an AI-driven on-screen cursor, alongside the user's, with proactive assistance capabilities as two core qualities. This design aims to guide users to navigate the on-screen tasks, assist them in ideation, and provide real-time, contextual feedback without requiring users to shift their attention away from the primary task interface.

To demonstrate Pointer Assistant, we conducted a comparative study to illustrate the interaction technique's capabilities. We selected personal financial budget planning as a benchmark task for the interaction technique. This task is chosen due to its screen-based spatial nature and the cognitive challenges posed during the task—such as what expense categories to consider, how much to budget, and how to adjust the resulting numbers to achieve certain financial goals. With these task characteristics, we expect the interaction technique's proactive suggestions, reminders, and comments generated using an LLM embedded in a pointer form to help guide users through the task, remind users about missing items, critique their input, and improve user satisfaction while reducing cognitive load compared to traditional chatbot interfaces.

Aspects of human-AI interaction that we investigate in our evaluative study for Pointer Assistant include user perception of the task (satisfaction with the process, the outcome, and task load), perceived "AI as a collaborator" qualities (perceived usefulness, cooperativity, social influence, trustworthiness), several agent characteristics (perceived proactiveness, intelligence, competence, seriousness, and dominance), and behavioral outcomes (task completion time, number of ideated budget categories, suggestion acceptance, and proportion of user's original categories). We hypothesize about each of the Pointer Assistant's key attributes that:

- **H1:** Chatbots with pointers will improve the above metrics including user perception of the task, perception of AI as a collaborator, agent characteristics, and behavioral metrics.
- **H2:** Proactive chatbots will improve the above metrics including user perception of the task, perception of AI as a collaborator, agent characteristics, and behavioral metrics.
- **H3:** Proactive chatbots with pointers will be the most effective condition for all of these measured outcome variables.

To answer these questions, we conducted a rigorous randomized controlled experiment involving 220 participants. We compared proactive and passive chatbot configurations, both with and without visual pointers, against a control condition lacking chatbot assistance. By examining these variations, we aimed to uncover the impact of chatbot behavior and visual guidance on the aforementioned key metrics regarding task perception, AI as a collaborator attributes, agent characteristics, and behavioral metrics.

Our study contributes to the growing body of knowledge on human-AI interaction in several key ways:

- We introduce and evaluate a novel interaction technique that combines AI-driven cursor guidance with proactive assistance for on-screen spatial tasks.
- We provide empirical evidence on the effects of pointer-based AI assistance and proactivity on user experience and task performance in the example task of financial planning.
- We offer insights into user perceptions and behaviors when interacting with an AI assistant that provides spatial guidance, informing future designs of human-AI collaborative interfaces.
- We discuss the implications of our findings for the broader field of AI-assisted interfaces, particularly in domains requiring complex decision-making and spatial task completion.

By exploring the intersection of AI assistance, proactivity, and visual cues, this research not only advances our understanding of effective human-AI collaboration but also provides practical guidelines for designing more intuitive and supportive AI assistants in digital applications.

2 Related Work

This work is situated within the following areas of research:

2.1 Designing Collaborative Interfaces for Human-AI Interaction

As AI systems have become increasingly prevalent in various domains, designing effective interfaces for human-AI collaboration has emerged as a critical research area. Prior work has explored different paradigms and approaches for facilitating productive interactions between humans and AI agents.

At the higher level, frameworks for developing a successful human-AI interaction were studied. Wang et al. [33] emphasized the importance of mutual goal understanding, preemptive task co-management, and shared progress tracking for effective human-AI collaboration. They advocate for incorporating perspectives from Computer-Supported Cooperative Work (CSCW) in designing AI systems to better integrate them into existing human workflows. This aligns with our approach of using a visual pointer to provide spatial guidance and real-time feedback. Cila [4] identified key qualities for designing human-agent collaborations, including code-of-conduct, task delegation, autonomy, intelligibility, common ground, offering help, and requesting help. These considerations guided our design's proactive and passive behaviors.

Several researchers have investigated specific interaction techniques to enhance human-AI collaboration. Quek and Petro [24] proposed the Human-Machine Perceptual Cooperation (HMPC) paradigm which combines human reasoning with machine perception to collaboratively solve spatio-perceptual intensive problems. Their work on shared perceptual load between humans and machines relates to our use of a proactive AI pointer to highlight relevant areas of the interface. In the context of task management, Toxtli et al. [31] explored using chatbots to mediate collaborative tasks within communication channels. Their insights on bot design for coordinating work inform our approach to integrating AI assistance directly into the user's workflow.

Moreover, researchers have studied various principles for effective human-AI collaboration. Subramonyam et al. [30] discussed how "leaky abstractions" can be utilized to bridge the gap between UX designers and AI engineers in collaborative system design. This concept of exposing low-level details across expertise boundaries informed our approach to making the AI's decision-making process more transparent through visual cues. Ehsan et al. [9] emphasized the importance of Human-Centered Explainable AI (HCXAI) in enhancing human-AI collaboration. They advocated for operationalizing holistic approaches to produce actionable frameworks and design guidelines. This perspective informed our focus on making the AI assistant's actions and reasoning more interpretable through visual cues and contextual feedback. Jacobsen et al. [13] studied perceived and measured task effectiveness in human-AI collaboration, finding that users both performed better and perceived themselves as more effective when working with AI assistance. These findings support our hypothesis that the Pointer Assistant can improve both actual and perceived performance in collaborative tasks.

The rise of Large Language Models (LLMs) in recent years has also prompted broader conversations about the nature of human-AI interactions. Traditionally, AI chatbots have been conceived as passive tools that simply comply with user commands. However, some scholars have proposed a more dynamic approach, suggesting that AI should be capable of taking a more engaged stance—such as being more proactive in providing feedback, asking Socratic questions to engage users' cognitive processes, giving critique, or even potentially challenging user inputs rather than acting as a simple, submissive chat interface [26]. Examples of this approach can be found in how Danry et al. reframed AI-generated explanations into a question form, resulting in improvements in the user's discernment of information presented [6], and how Cvetkovic et al. showed that AI-generated critique can help improve the quality of ideas in brainstorming tasks [5].

By building on these diverse approaches to human-AI collaboration, our work on the Pointer Assistant aims to create a more intuitive and effective interface for AI-assisted on-screen tasks. We incorporate elements of visual guidance and proactive assistance to enhance the collaborative experience between users and AI systems.

2.2 Mouse, Cursor, and Pointing Devices

Since the introduction of graphical user interfaces, pointing devices have played a crucial role in human-computer interaction. The computer mouse, introduced by Engelbart in 1964, has become the de facto standard for cursor control [14]. However, researchers have continually explored alternative pointing devices and interaction techniques to improve efficiency, ergonomics, and accessibility.

Several studies have compared the performance of different pointing devices. MacKenzie et al. conducted a seminal study comparing a mouse, trackball, and stylus with a tablet for pointing and dragging tasks [19]. They found that the stylus outperformed the mouse in pointing tasks, while the mouse was superior for dragging. Cao et al. later compared single-finger, whole-hand, and hybrid pointing devices, demonstrating the potential of combining finger and hand movements for improved pointing performance [3].

The orientation and design of cursors have also been subjects of research. Po et al. investigated the effects of cursor orientations on mouse, pointer, and pen interactions, finding that orientation-neutral cursors generally performed best, especially for pointer interactions [23]. Cursor design innovations include the multifunctional cursor proposed by Muller, which overlays multiple operation icons into the cursor image to improve interface efficiency [21]. These works inspired our use of different animated hand gesture icons and emojis as "emotes" to convey the chatbot's expressions.

Researchers have also explored novel interaction techniques to enhance pointing performance. Blanch and Ortega introduced the rake cursor technique, which combines mouse input with gaze tracking to significantly outperform traditional mouse-only pointing [2]. Endert et al. developed ChairMouse, leveraging natural chair rotation for large-scale cursor movement on high-resolution displays [10].

For users with disabilities, alternative pointing devices and techniques have been crucial. Harada et al. conducted a longitudinal study on people learning to use continuous voice-based cursor control, demonstrating its potential as an accessible input method [12]. Similarly, eye-gaze tracking has been explored as an alternative pointing method, with Kumar et al. proposing EyePoint, a technique combining gaze input with keyboard triggers for efficient pointing and selection [16].

Recent advancements in technology have led to the exploration of more sophisticated pointing devices. Zhou et al. introduced the Depth-Adaptive Cursor, a technique for integrating 2D mouse input into 3D virtual reality environments, addressing challenges such as binocular rivalry and perspective mismatch [34]. This research highlights the ongoing efforts to adapt traditional pointing devices to new computing paradigms. The integration of haptic feedback into pointing devices has also been an area of active research. Rosenberg et al. developed a force feedback mouse interface to provide enhanced cursor control through tactile sensations [25]. This approach aimed to reduce the disconnect between visual and physical interactions in graphical user interfaces.

Developments over the past decade in consumer applications have also been inspired by the prevalence of pointing in graphical user interfaces. The rise in popularity of real-time online collaboration software such as Miro [20] and Figma [11] in recent years has made users become familiar with multiple users being on the same "board" editing the same file at the same time. These applications display other people's cursor movements and actions in real time, fostering the feeling of presence and collaboration between the users. Pointer Assistant was inspired by these real-time human-to-human interactions.

3 System Design

This paper presents a new interaction technique between users and large language models (LLMs) that we will call Pointer Assistant. This system involves two key features: 1) *Pointer Form*, where the LLM takes the form of another mouse cursor alongside the user's with chat a bubble attached to the cursor instead of having a dedicated chat log area, and 2) *Proactivity*, the AI's ability to proactively send messages to the user based on on-screen user activities.

3.1 Interaction Description

Pointer Assistant is designed to assist users with on-screen tasks, especially spatial ones, by giving real-time suggestions while pointing to relevant areas on the screen. This can help guide the users to navigate a task, ideate different inputs for a task, and give feedback on their input. This design was inspired by human-to-human "cursor chat" interactions in real time collaborative design applications such as Figma [11].

The Pointer Assistant interaction technique seeks to improve on the popular human-LLM interaction technique of having a dedicated chat interface with stacked chat bubbles in the following ways. First, it aims to help users stay focused on the task without having to context-switch between different user interface panels when they need to use an LLM, making the interaction more seamless. This characteristic is expected to consequently lower the task's cognitive load and time to complete the task. Second, it also aims to provide users with AI-generated real-time feedback on their tasks. This may include guidance, critiques, questions, and reminders, which could result in better task outcomes. For example, with the help of Pointer Assistant, users may be able to come up with more and higher-quality ideas in brainstorming tasks, or more well-rounded and well-structured arguments in writing tasks. The guidance given by the AI pointer may also make it easier for users to navigate complex interfaces, potentially improving the users's satisfaction with the experience while also lowering cognitive load. Finally, these improvements are expected to contribute to improvements in the overall user experience of using an LLM chatbot.

The details of this design, illustrated in (Fig. 1), are as follows: upon enabling Pointer Assistant, users will notice an additional AI-powered mouse pointer on the screen alongside their own. For this prototype, the user's and AI's pointers are in orange and green, respectively. When the user acts on the screen, e.g., inputting data into a text box, the LLM-powered pointer will proactively give them comments about their input or suggestions for what to do next. These messages will be gradually displayed next to the AI's pointer and disappear after a few seconds. While displaying the message, the AI pointer will also move to point at the area in question to help highlight the relevant part of the screen while also changing its appearance to animated hand-shaped emojis to depict hand gestures, accompanying the message. These hand gestures, or "emotes", include *pointing*, *waving*, *thumbs up*, *OK*, and "*I love you*" (as shown in Fig. 4). These emotes are designed to make Pointer Assistant more expressive and feel less serious. Pointer Assistant will respond to the user's actions after they have stopped clicking or typing for 0.5-1 second.

Additionally, users can also send messages directly to Pointer Assistant by typing on an empty area anywhere on the screen and pressing the 'enter' or 'return' key on the keyboard to send the message (Fig. 1, right). In this case, the Pointer Assistant will also respond directly to the user's chat message like a normal chatbot albeit with a different presentation of chat messages as described above. Besides direct interaction with the user, the AI pointer will also occasionally move to different areas of the screen at random time intervals while idling to make the cursor appear more lively.

The reason Pointer Assistant is designed as an additional pointer is to leverage users' existing mental model of mouse pointers for pointing at important elements on the screen, guiding users' focus to one area of interest at a time. Moving the message boxes closer to the user's and the chatbot's cursors is also aimed at reducing the mental load that occurs when users need to look away from the task to interact with an LLM. Moreover, these chat messages are also designed to be transient, disappearing after a few seconds, to make the interaction feel more like a verbal conversation rather than transactional. This is expected to potentially result in users perceiving the interaction as less formal and serious and ultimately feel more like a collaborator, rather than a third party that the user consults besides the main task.

3.2 Potential Use Cases

Pointer Assistant is designed for spatial tasks where real-time proactive feedback or guidance may be helpful to the user. In other words, tasks that may benefit from this interaction technique should exhibit these key characteristics:

- *On-Screen Spatiality*: The task involves multiple parts of the screen where the users' attention frequently shifts between different elements, such as navigating multiple tables, lines of text, user interface panels, etc. In these tasks, users may overlook certain elements that require attention. Pointer Assistant's ability to move and point aims to help focus users' attention on specific areas instead of having to locate them themselves, potentially reducing cognitive load.

- *Opportunities for Real-Time Feedback/Guidance*: The task is one where users can benefit from a "second pair of eyes" that continuously review their work and provide suggestions and/or corrective feedback. Such tasks pose a challenge in helping users quickly spot areas of improvement in their work. Pointer Assistant's proactive comments, based on the user's on-screen actions, offer this type of assistance.
- *Small, Actionable Steps*: Pointer Assistant's design emphasizes short, conversational exchanges, making it most effective for tasks that can be broken down into small, actionable steps (e.g., filling a text box, fixing a sentence, clicking a button). Unlike long instructions that can be difficult to follow, Pointer Assistant's concise step-by-step, and in-context interaction style helps users focus on one action at a time.

In these tasks, the assistant will act as a virtual task facilitator along three key aspects: 1) guiding the users to navigate the interface, 2) assisting users with ideation, and 3) giving feedback to their inputs. With these characteristics, we envision Pointer Assistant to be suitable for tasks such as writing, brainstorming, and planning in multiple domains such as design, programming, education, and finance, as illustrated through screen mock-ups in Fig 2. However, while Pointer Assistant is well-suited for these types of tasks, its effectiveness may be limited in situations requiring extensive long-form dialogue with the AI assistant, or tasks where real-time feedback is less critical.

An example of a task that features these characteristics is programming (Fig. 2a). As opposed to the traditional programmer-AI workflow of writing prompts to ask an LLM model for a code snippet or a code review, Pointer Assistant can help users point out potential errors in the code in real time. While this can also be done with annotation-based interaction, having an extra AI pointer can help users focus on one step at a time. Moreover, if needed, users can easily ask further clarifying questions directly to the assistant in a natural way through the cursor chat feature.

Another potential use case is in design brainstorming (Fig. 2b). With Pointer Assistant's proactive suggestions and comments, the assistant can help users come up with more ideas by nudging them through the chat messages. Users can also ask for more specific ideas via the cursor chat interface directly on the workspace without having to switch to a separate chat interface, potentially making the task feel more seamless.

Education use cases can also benefit especially from the guidance aspects of the interaction technique. Pointer Assistant can provide step-by-step guidance and real-time comments while learners complete a task such as solving a physics problem (Fig. 2c). The use of Pointer Assistant in this context broadens the possibilities of personalized learning by providing guidance at the learner's pace. Moreover, the ability for learners to ask the model to clarify its comments, or pose their own relevant questions may also help learners understand the lesson more deeply.

These potential use cases demonstrate how Pointer Assistant can provide context-specific task facilitation across diverse domains. By enabling proactive, localized feedback, this approach offers a more integrated and responsive way of collaborating with AI assistants.

4 System Implementation

4.1 Example Task

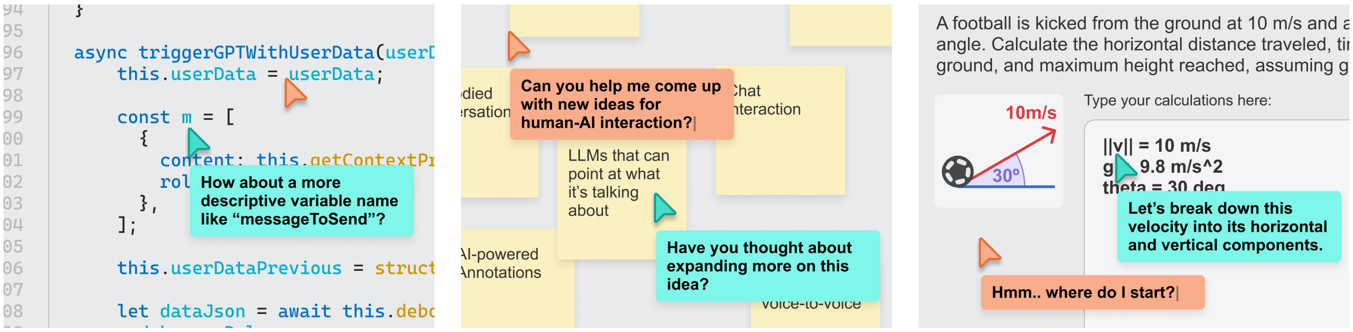
Financial planning is one of the tasks that have been studied with different kinds of computerized assistance used to enhance the user's information processing abilities [8, 17, 27, 29]. Within this context of HCI intervention in financial planning, we also see the potential of applying Pointer Assistant to such tasks.

For this study, we employed annual financial budget planning as a pilot task to highlight Pointer Assistant's abilities. Adapted from the zero-based budgeting technique in personal finance [28], the task requires planners to allocate every dollar of their income into budget categories and planned savings. In other words, planners come up with a list of all expense categories they are planning to spend in a year. Each category is then assigned an approximate monthly or annual amount in their currency until the total expenses become equal to their projected income. This list is then used by the planner as their budget for the year.

This task is selected as our pilot task because it highlights the benefits of the Pointer Assistant technique in a well-rounded way, demonstrating the three main aspects of guidance, ideation, and real-time feedback. The task is also spatial in nature and requires a considerable amount of user input; users have to navigate through multiple numbers in different tables on the screen. This can highlight how the Pointer Assistant's pointer form can help guide users step-by-step through complex interfaces, reducing cognitive load. What's more, the task presents cognitive challenges that could benefit from real-time feedback and reminders as stimuli for ideation. For instance, it requires users to come up with a number of expense categories, some of which may have been frequently overlooked suppose the users have to list them on their own without assistance. Moreover, users also have to come up with an estimation for the budget amount planned for each category and make decisions on how to adjust each category to meet their savings goals. These are examples of areas in which we expect Pointer Assistant's proactive real-time feedback capabilities to help users come up with a more comprehensive and realistic budget plan. Additionally, in terms of implementation, the structured, table-based inputs featured in this task also help simplify our cursor location mapping technique, reducing the risk of user distraction due to imprecise location mapping of the AI cursor.

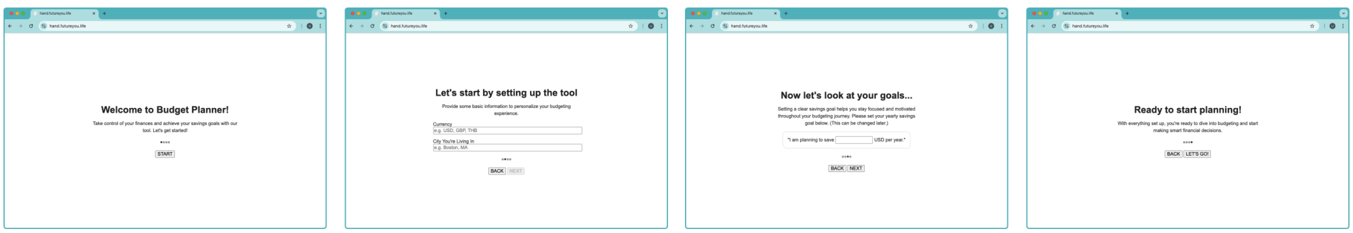
For this task, we hypothesized that the Pointer Assistant interaction technique would help users come up with more categories, covering more overlooked expenses with higher satisfaction for the resulting budget plans while reducing task load when compared to traditional chat interfaces.

The task is implemented into our prototype with the user flow as illustrated in Fig. 3. First, users are asked about their currency and the city they are living in as an initial context for the LLM. Then, they are asked about their savings goal ("I am planning to save ____ [currency] this year") (Fig. 3a –3d). After these initial steps, users are shown instructions to 1) list monthly expenses, 2) list less frequent expenses, 3) adjust the budget or savings goal, and 4) finalize and save their plan (Fig. 3e). Next, the system will introduce Pointer

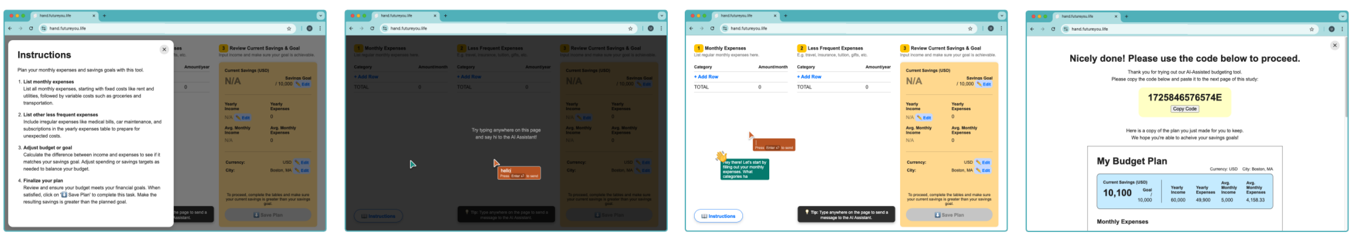


(a) Coding: Virtual Pair Programmer **(b) Brainstorming: Ideas Companion** **(c) Education: Personal Tutor**

Figure 2: Screen mock-ups for example potential use cases of Pointer Assistant in various domains.



(a) Greeting screen **(b) Currency and city input** **(c) Savings goal setting** **(d) Ready to start**



(e) Instructions **(f) Pointer Assistant tutorial** **(g) Main budgeting screen** **(h) Budget plan summary**

Figure 3: Overall user flow of the budget planning task with Pointer Assistant (user's and AI's pointers are in orange and green, respectively.)

Assistant and instruct the user to “say hi” to it. This step serves as a quick tutorial for the on-screen chat system. Once they have greeted the AI pointer, they will see the main budgeting screen as shown in Fig. 3g. The screen features three sections: 1) *monthly expenses*, where users can add their monthly expenses categories, e.g., food, rent, shopping, and transportation, and their monthly budgeted amounts, 2) *less frequent expenses*, which is another table for yearly or semi-yearly expenses such as insurance, car maintenance, or travel, and 3) *review*, showing the summary of their budget and the resulting amount they can save based on their written budget. The third section will be updated in real time according to the changes in the two budget tables.

Throughout the task, Pointer Assistant will guide the users to perform each step of the task by pointing at relevant areas of the screen and giving brief instructions through its cursor-attached chat bubble. Moreover, it will continuously help users brainstorm budget categories and give feedback on the amount. For example, the assistant may suggest users add groceries, subscriptions, car maintenance, etc. to their budget. It can also proactively point out possible errors and typos or questionable inputs that the user has made in the expense tables. Users may also ask the assistant for help in areas such as coming up with an appropriate budget amount for a category, commenting on their budget amount, providing more budget category ideas, or even just chatting with the AI assistant.

After both tables have enough details, they are then asked by the AI pointer to fill in their annual income in the third section which will then reveal the amount they can save based on their specified budget and their income. If the amount is less than the initially set goal, Pointer Assistant will suggest the user adjust their budget or savings goal and help suggest where to make changes. Otherwise, it will instruct the user to click the “Save Plan” button to complete the task. The button is only enabled if they have filled their expense items, annual income, and the remaining amount for savings, calculated as $AnnualIncome - (MonthlyExpenses \times 12 +$

LessFrequentExpenses), is greater than or equal to their set savings goal. Once completed, the system will display the user's finished budget as a summary for the user.

4.2 Technical Implementation Details

Pointer Assistant and the budget planning task were implemented as a web application designed for desktop or laptop computers. Pointer Assistant is powered by a large language model (LLM) through OpenAI's ChatGPT-4o chat completion API. The application-LLM interaction (Fig. 4) is activated by two triggers: 1) user actions such as filling or adding a new row to an expense table (with a 500-millisecond debouncing mechanism, i.e. 500 milliseconds after the user has stopped clicking or typing), and 2) when the user sends a message directly to Pointer Assistant. Once an action is triggered, a message is sent to the LLM for a JSON-formatted response.

Then, the user's workspace data, i.e., budget tables and other user inputs (e.g., saving goals) along with relevant computed values (e.g., resulting savings amount) are serialized into JSON format. The table-to-JSON conversion converts each on-screen table (as can be seen in Fig. 3g) into an array of objects with the keys "`Category`," and "`Amount/month`" or "`Amount/year`," reflecting each table's column headers.

Next, the compiled prompt is sent to the LLM with each request message containing multiple parts as follows:

(1) A pre-prompt that includes instructions for the model to assume the role of a personal finance assistant, to give feedback on user input, to help the user brainstorm possible categories they might have missed, to keep its responses short and conversational, and the steps in the flow of the budgeting task as mentioned in the previous section. This pre-prompt also includes the user's context including their income, currency, and city.
(2) The user's serialized JSON data as described. Both the current and the previous states of the workspace are included to make the model able to comment specifically on what has been changed between each interaction turn.
(3) The instructions for the model to respond in JSON format with keys including the response message, the choice of emoji hand gestures it wants to express along with the message, and a reference string for the keys in the user input JSON that the model determines as relevant to its response.
(4) In the case where the interaction turn is initiated by the user's typed message, the user's message is also appended to the prompt.

Additionally, to minimize the length of the message sent each time to the model while maintaining the task's current state, it is also prompted to include summaries of each interaction with the user to its response. Specifically, the model is prompted to summarize its interactions with the user from the beginning until the latest interaction including what the user said and what it said to the user, while also taking the previous summary into account. With this prompt, the model will respond with a summary such as: *"The user greeted me, and I suggested starting with filling in their monthly expenses, specifically categories like rent, utilities, groceries, and transportation. The user asked for suggestions, and I guided them to fill those into their budget planning table. The user has now added 'rent', 'utilities', 'food', and 'transportation' as categories with specified amounts but left one blank. I encouraged them to consider entertainment or insurance next."* This summary is then fed back with each subsequent request to the model as a brief history of the whole session up until that point instead of having to send the entire chat and interaction history to the model.

Once the response message arrives from the model, 1) the chat message is displayed at the assistant's pointer, 2) the pointer "emote" specified in the response is performed, and 3) the returned reference JSON key is then mapped to corresponding screen coordinates where the pointer then moves to. This conversion is possible since each main HTML elements are marked with the same identifier as the key used in the user context JSON. For instance, the identifier of the input element for income ("`income`") is the same as the JSON key containing the user's income data. When the model responds with the identifier, the user interface system searches for an HTML element with the same identifier. The matching element's bounding box is then used to determine the target pointer coordinates. If the model refers to a piece of user data nested deeper within the JSON structure, i.e. data located in tables, it will return a more detailed reference key (e.g., "`MonthlyExpenses[0].Amount/month`"). This more detailed key is then parsed into parts and matched with the appropriate table cell. In this example, the system will look for the HTML table with the identifier "`MonthlyExpenses`," then its first row ("`[0]`"), then the cell corresponding to the "Amount/month" column. The located cell is then used as the target location to which the pointer moves.

4.3 Implementing Pointer Assistant in Other User Cases

In order to adapt Pointer Assistant to other use cases, three modular parts of this system need to be edited as denoted in Fig. 4. These include serialization of the on-screen workspace data during pre-processing (step 2), the basic pre-prompt of the system describing the task to the LLM (step 3), and the mapping method that maps the reference key back to the corresponding UI element (step 5).

The basic instructions pre-prompt in step 3 can simply be changed to describe the new task. For example, to use Pointer Assistant as a virtual pair programmer, this part may include instructions for the model to assume the role of a code reviewer, given the task of spotting potential errors and code quality issues. Alternatively, in tutoring tasks, it may be prompted to act as a tutor that walks the user through solving a given problem step by step.

For data serialization and the mapping of reference keys to UI locations, there are requirements that 1) the user's workspace data must be serializable into a JSON string without losing important context, and 2) each user input that Pointer Assistant is expected to be able to point at must be addressable via the keys of the JSON.

For example, in writing-heavy tasks such as coding or educational exercises (as illustrated above), the user's writing can be serialized as an array with each line of text as an array item:

```
{ content: [ "text of line 1", ... ] }
```

This serialized data is accompanied in the prompt by a description of the data format (e.g., "each array item represents each line of code"). In such tasks, the LLM prompt should also include further instructions for the model to respond with the line number and the

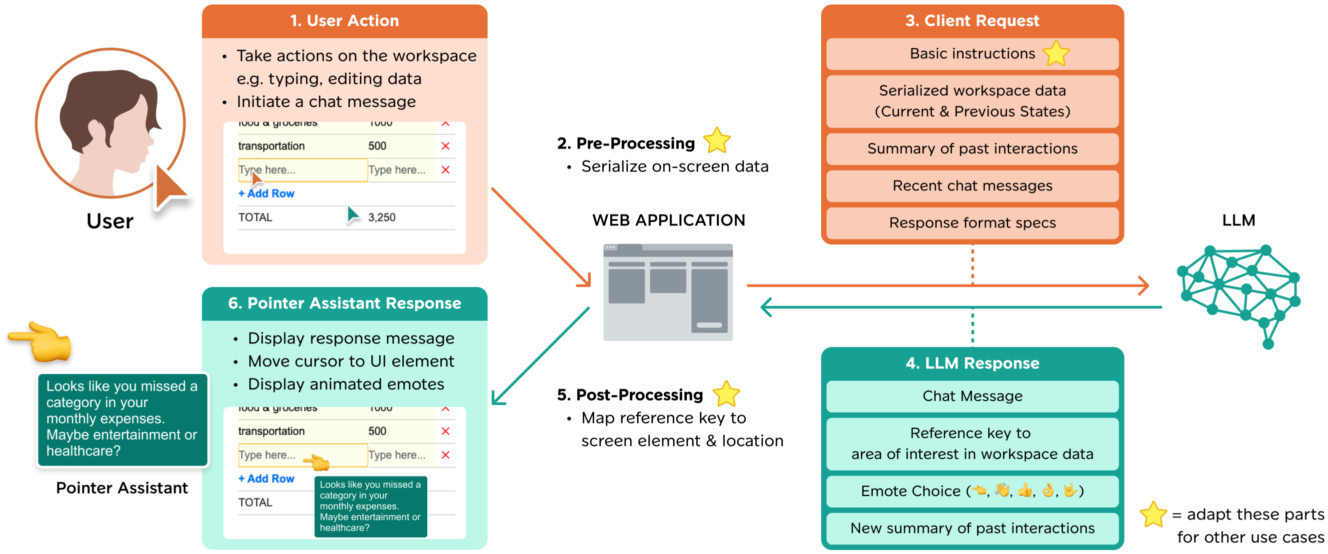


Figure 4: Interaction details including request and response message structures between the application and the LLM

exact excerpt within the line it is referring to for a more granular location matching:

```
{ "lineNumber": 1, "excerpt": "i = 0" }
```

This return data can then be used to locate the on-screen coordinates of the excerpt of interest during the post-processing step. On the other hand, for brainstorming tasks with sticky notes where the notes' clustering may also convey important context, including the grouping structure into the serialized data would enable the model to better "understand" the task's context, leading to more relevant and helpful responses. For instance, the on-screen data in this task may be serialized to:

```
{
  notes: [ { "id": 0, "content": "note 1" }, ... ],
  groupingByIds: [ [ 0, 1, 3 ], ... ]
}
```

By adapting these modular parts, one can apply Pointer Assistant to other use cases to enable AI-generated proactive and spatial feedback for their users.

5 Exploratory Study Design

An exploratory empirical study is conducted to assess the effectiveness of the Pointer Assistant interaction technique in assisting users with the aforementioned financial planning task.

5.1 Study Methodology

Since Pointer Assistant features two core unique characteristics compared to traditional LLM-based chatbots, namely, cursor form and proactivity, we designed a 2×2 randomized controlled study that employs a between-subjects experimental design. An extra passive control condition where the user performs the budgeting task without any AI assistance was also added to the comparison as a baseline, resulting in a total of five conditions:

(1) *No Chatbot* (passive control): the participants complete the budget planning task without any chatbot (Fig. 5a)
(2) *Passive Chatbot without Pointer* (active control): the assistant is displayed as a traditional chat interface on the side of the screen (Fig. 5b) and will respond only to chat messages initiated by the user.
(3) *Proactive Chatbot without Pointer*: the assistant is displayed as a traditional chat interface and would proactively respond to the user's actions on the budget tables.
(4) *Passive Chatbot with Pointer*: the assistant is displayed in pointer form (Fig. 5c) but would not comment on the user's input unless asked to.
(5) *Proactive Chatbot with Pointer*: Pointer Assistant is fully employed as described in previous sections.

Upon beginning the study, participants were informed about the study protocol and gave their consent to participate in the study. Then, they were randomly assigned to one of these five conditions to complete a web-based financial planning activity as described in the previous section.

Chatbots of different characteristics, if any, were enabled based on the participant's assigned experimental condition (Fig. 5). Behavioral measurements including task completion time, number of categories covered, suggestion acceptance, and proportion of original categories were recorded throughout the task. Suggestion acceptance was calculated as the proportion between the number of AI's suggestions each participant put into their expense tables and the total number of AI's suggestions for the participant. The list of AI's suggestions was extracted from the chat history recorded during the sessions. The participants' inputs with the same meaning as an AI's suggestion, even when they did not exactly match in string comparison (e.g. "electric" vs "electricity bills", or typos such as "car maintence" vs "car maintenance") were counted as an accepted suggestion. On the other hand, the proportion of original categories was measured as the number of a participant's budget categories that were not suggested by the AI assistant, divided by the total number of categories the participant put into their tables. Suggestion acceptance and the proportion of original categories were not measured in the control condition as there were no AI suggestions. Completion time and number of categories were measured to assess Pointer Assistant's design goals regarding user's task performance while suggestion acceptance and proportion of original categories were measured to investigate potential side effects of over-reliance on different configurations of AI chatbots.

Once they had completed the task, participants responded to a survey measuring self-report metrics to assess Pointer Assistant's performance in relation to its design goals of reduced perceived task load, improved user experience, and satisfaction with the outcome, as well as other relevant measurements assessing user perception towards the chatbot. The metrics in these categories, adapted from validated scales, were included in the survey:

- *Task Perception*: satisfaction with the experience ("I was satisfied with the process of working through the task."), satisfaction with the results ("I was satisfied with the outcome of the task."), and task load scale [15]
- *AI as a Collaborator*: usefulness scale [7], perceived cooperativity scale [1], social influence scales (positive and negative) [15], and trustworthiness scales (positive and negative) [15]
- *Agent Characteristics* [32]: passive–proactive, incompetent–competent, unintelligent–intelligent, playful–serious, and submissive–dominant scales

These self-report scales were posed to the participants as 7-point Likert scale questions. Agent characteristics scales were rated using a 100-point scale. Note that the control condition (no AI Chatbot) only completed the subscales in the task perception category. Free-text qualitative answers about the participant's general experience interacting with the assistant were also collected.

This experiment and survey were conducted in English through online, unmoderated sessions. Participants performed the task on their desktop/laptop computers through a web browser via the interface embedded in a Qualtrics survey, where they also completed the post-survey. Each session took 10-20 minutes to complete in total with the financial planning task taking roughly 10 minutes to complete on average.

5.2 Participants

A total of 220 participants were recruited for this study, resulting in 42-48 participants per condition after the random assignment. The participants were recruited through the online participant recruitment platform CloudResearch and compensated upon the study's completion. Participants included United States-based adults over 18 years old. The vast majority of participants (78.6%) were under 45 years old, with nearly half (47.7%) younger than 35. Participants' educational backgrounds ranged from high school to graduate/professional degrees, with 60% holding a bachelor's degree or higher. Participants who reported experiencing technical issues, failing to complete the study, or failing attention checks were excluded from the analysis.

5.3 Data Analysis

Data analysis was conducted by first calculating average scores for each subscale resulting in a total of 18 measurements for each participant. Analysis of Variance (ANOVA) models were employed to examine differences between groups for each outcome variable. Statistically significant ANOVA groups were then followed up with Tukey's post-hoc tests to further investigate significant pairwise differences between conditions.

Additionally, thanks to the 2×2 study design, we also further analyzed the results with multiple linear regression with the terms "*Pointer*" and "*Proactive*" representing the interventions, and the interaction term "*Pointer* ∗ *Proactive*." Regression coefficients β were used to determine important factors that serve as predictors of each measured metric.

6 Experimental Results

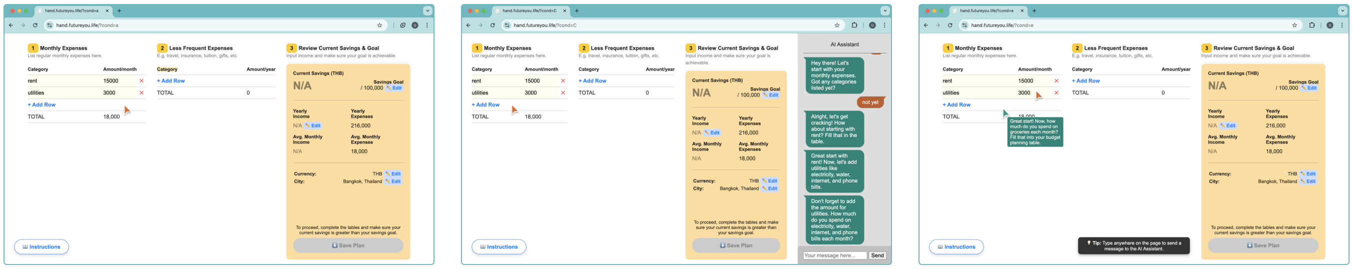
This study aims to measure the effectiveness of the Pointer Assistant interaction technique and potentially explore other prominent characteristics resulting from this interaction design. This is done by breaking the design down into two core features; pointer form and proactivity, leading to the 2×2 study design with an additional control condition, totaling five experimental conditions. The dependent variables included the aforementioned behavioral metrics and subscales of Task Perception Metrics, AI as a Collaborator Metrics, and Agent Characteristics Metrics.

6.1 Measured Mean, Standard Deviation, and ANOVA

The measured mean and standard deviation values from the study with five distinct conditions were presented in Tables 1–3 and Fig. 6 along with p-values from ANOVA. The analysis indicated significant results in the number of filled categories ($p = 0.0294$), the proportion of original categories ($p =< 0.0001$), satisfaction with the experience ($p = 0.0156$), task load ($p = 0.0029$), proactiveness ($p = 0.0003$), and dominance ($p = 0.0011$). No significant differences were found in other metrics ($p > 0.05$). Post-hoc tests further reflected significant differences between condition pairs in the following metrics groups:

6.1.1 Behavioral Metrics. First, significant effects in the **proportion of original categories** were found between all pairs of the four AI-enabled conditions. The proportion in *passive chatbot with pointer* condition was significantly higher than in the conditions of *passive chatbot without pointer* ($p = 0.0138$), *proactive chatbot without pointer* ($p < 0.0001$), and *proactive chatbot with pointer* ($p < 0.0001$). The metric measured in *passive chatbot without pointer* was also higher than *proactive chatbot without pointer* ($p < 0.0001$), and *proactive chatbot with pointer* ($p = 0.0008$). Furthermore, *proactive chatbot with pointer*'s metric was higher than that of *proactive chatbot without pointer* ($p = 0.0487$). In other words, the proportion of original categories was higher in passive and pointer conditions compared to their proactive and 'no pointer' counterparts, respectively. No significant pairwise differences were found in **number of filled categories** despite significant ANOVA results.

6.1.2 Task Perception Metrics. The results also indicated that **satisfaction with the experience** decreased in the *passive chatbot*



(a) No chatbot (b) "Without pointer" conditions (c) "With pointer" conditions

Figure 5: Screenshots of the chatbot configuration for each set of the experimental conditions.

without pointer compared to the *no chatbot* condition ($p = 0.0270$). **Perceived task load** increased in *proactive chatbot without pointer* compared to *no chatbot* condition ($p = 0.0407$), and decreased in *passive chatbot with pointer* condition compared to *proactive chatbot without pointer* ($p = 0.0033$). There was also a notable significant decrease in task load between *proactive chatbot with pointer* from *proactive chatbot without pointer* ($p = 0.0485$).

6.1.3 Agent Characteristics Metrics and AI as a Collaborator Metrics. For agent characteristic subscales, the results showed that the **Passive-Proactive** rating for the *proactive chatbot without pointer* was greater than in the *passive chatbot with pointer* ($p = 0.0171$) and *passive chatbot without pointer* ($p = 0.0387$) conditions. The Perceived proactiveness in the *proactive chatbot with pointer* was also greater than in the *passive chatbot with pointer* ($p = 0.0029$) and *passive chatbot without pointer* ($p = 0.0073$) conditions. In other words, all *proactive* conditions were expectedly perceived as significantly more proactive than their passive counterparts.

Finally, on the **Submissive-Dominant** scale, *proactive chatbot without pointer* was rated higher than *passive chatbot with pointer* ($p = 0.0349$), and *proactive chatbot with pointer* was also rated higher than *passive chatbot with pointer* ($p = 0.0029$), and *passive chatbot without pointer* ($p = 0.0219$). Pairwise comparisons between other metrics were not statistically significant. Additionally, no significant differences were found in AI as a Collaborator metrics.

6.2 Multiple Linear Regression Analysis

Results from multiple linear regression analysis (Tables 4–6 and Fig. 7) further highlighted the key characteristics that drive each measurement. Statistically significant regression coefficients indicated each intervention term as positive and negative predictors for the accompanying dependent variables.

For behavioral metrics, there was a significant trend toward the *Proactive* term being a positive predictor of the **number of filled categories** ($\beta = 1.614, p = 0.047$). The **proportion of original categories** filled by the user was also significantly affected by both *Pointer* ($\beta = 0.168, p = 0.003$) and *Proactive* ($\beta = -0.358, p < 0.001$) terms with the terms being positive and negative predictors, respectively.

For task perception metrics, it was found that the *Pointer* intervention term was a significant positive predictor of **satisfaction with the experience** ($\beta = 0.554, p = 0.042$), and a significant negative predictor of **perceived task load** ($\beta = -0.511, p = 0.020$) (*Pointer* intervention predicts reduced perceived task load). The *Pointer* term was a near significant positive predictor for **user's satisfaction with the result of the task** ($\beta = 0.460, p = 0.056$).

In agent characteristics metrics, the *Proactive* intervention term was found to be a significant positive predictor for the **passive-proactive** ($\beta = 14.318, p = 0.008$) and **submissive-dominant** ($\beta = 8.864, p = 0.040$) ratings. No significant effects between other intervention terms and dependent variables were found ($p > 0.05$).

6.3 Qualitative Responses

Certain themes in the qualitative data from the free-text general comments responses further complement the quantitative metrics. In general, participants in all chatbot-enabled (non-control) conditions commented that the assistant helped remind them of categories they might have forgotten. This comment was especially prevalent for the proactive conditions. For example, some participants in the *proactive chatbot with pointer* condition noted about their experience that *"It was very positive. I love how it was giving me ideas even before I was able to think of them,"* and *"When it was recommending categories, it was useful."* The proactive chatbot was also commented positively on its ability to help correct user errors (*"I felt that the chatbox was particularly helpful in catching an error I made in typing the amount. I had entered an extra zero, which would have made a huge difference but the chatbox caught it, which was quite clever on the chatbot part,"* and *"I loved it. ... I think it was when I wanted to put transportation and dining under less frequent expenses and it called my attention to it and let me know those would be better under monthly expenses."*) On the other hand, the proactive chatbot was also perceived by some participants as pushy and annoying (*"... but then he started nagging and going over things too often."*, and *"It was very persistent and it felt strange having to explain why I wasn't putting a topic in a row it kept pushing me to put in."*) It was also reported that the chatbot tended to repeat what it had said to the user. Inversely, some participants in the passive chatbot conditions noted that they did not interact a lot with the chatbot and mostly did the budgeting task themselves (*"There was not much for the chatbot to do since I was just inputting my money information,"* (*passive chatbot with pointer*) and *"I paid more attention to filling out my information than interacting with the chatbot."* (*passive chatbot without pointer*)).

The design of the chatbot as an on-screen pointer was generally perceived as helpful visual guidance. For instance, participants in these conditions noted *"I like that I could type anywhere and then*

Table 1: Measured mean and (standard deviation) for each behavioral and task perception metric with p-value from ANOVA: significant differences between experimental conditions were found in number of filled categories, proportion of original categories, satisfaction with the experience, and perceived task load.

Condition	Behavioral Metrics				Task Perception (1-7 Likert)		
	Duration (seconds)	# of Filled Categories	Suggestion Acceptance (0-1)	Prop. Orig. Categories (0-1)	Satisfaction w/ the Result	Satisfaction w/ the Experience	Task Load
No chatbot (control)	414.06 (324.35)	8.81 (2.80)	n/a	n/a	6.19 (0.82)	6.35 (0.70)	2.46 (1.06)
Passive w/o Pointer	564.99 (420.62)	9.07 (3.32)	0.40 (0.19)	0.71 (0.29)	5.66 (1.18)	5.64 (1.31)	2.77 (0.99)
Proactive w/o Pointer	577.45 (358.38)	10.68 (4.25)	0.38 (0.16)	0.35 (0.24)	5.73 (1.28)	5.73 (1.47)	3.06 (1.04)
Passive w/ Pointer	483.17 (449.16)	9.26 (3.46)	0.41 (0.34)	0.88 (0.21)	6.12 (0.63)	6.19 (0.80)	2.26 (0.85)
Proactive w/ Pointer	651.18 (610.66)	10.62 (3.99)	0.32 (0.15)	0.50 (0.27)	5.86 (1.20)	5.90 (1.32)	2.45 (1.14)
ANOVA p-value	0.1054	0.0294*	0.2683	<0.0001***	0.0661	0.0156*	0.0029**

*p<0.05, **p<0.01, ***p<0.001

Table 2: Measured mean and (standard deviation) for "AI as a Collaborator" metrics, ratings from 1-7 Likert scale: no significant differences were found between experimental conditions.

Condition	Usefulness	Cooperativity	Social Influence (Positive)	Social Influence (Negative)	Trustworthiness (Positive)	Trustworthiness (Negative)
Passive w/o Pointer	4.36 (1.56)	4.64 (1.01)	3.68 (1.75)	3.57 (1.16)	4.69 (1.22)	3.03 (1.32)
Proactive w/o Pointer	4.75 (1.64)	4.87 (1.18)	4.08 (1.79)	3.81 (1.27)	4.75 (1.48)	3.16 (1.65)
Passive w/ Pointer	4.76 (1.51)	4.74 (1.25)	3.24 (1.74)	3.42 (0.88)	4.80 (1.35)	2.51 (1.38)
Proactive w/ Pointer	4.55 (1.68)	4.82 (1.22)	3.76 (1.66)	3.79 (1.28)	4.70 (1.57)	2.89 (1.75)
ANOVA p-value	0.5983	0.8163	0.1670	0.3513	0.9853	0.2366

*p<0.05, **p<0.01, ***p<0.001

Table 3: Measured mean and (standard deviation) for "Agent Characteristics" metrics, ratings from 0-100 Likert scale: significant differences between experimental conditions were found in the passive-proactive and the submissive-dominant subscales.

Condition	Passive (0) –Proactive (100)	Incompetent (0) –Competent (100)	Unintelligent (0) –Intelligent (100)	Playful (0) –Serious (100)	Submissive (0) –Dominant (100)
Passive w/o Pointer	58.64 (23.19)	69.32 (19.81)	67.95 (21.84)	62.27 (27.52)	48.18 (19.44)
Proactive w/o Pointer	72.95 (24.36)	69.09 (26.66)	66.59 (26.05)	62.95 (24.93)	57.05 (20.75)
Passive w/ Pointer	56.90 (30.48)	70.95 (23.67)	69.05 (23.46)	59.76 (23.00)	45.24 (19.28)
Proactive w/ Pointer	76.19 (20.83)	73.33 (25.44)	70.00 (26.04)	61.43 (28.59)	60.71 (20.65)
ANOVA p-value	0.0003***	0.8371	0.9261	0.9487	0.0011**

*p<0.05, **p<0.01, ***p<0.001

the chatbot would move the mouse to where I had to go. I was a bit confused at first why there were 2 mouses on the screen but I quickly understood," and *"[It] gave me cues to make the job easier. It would point the next steps out."* However, the pointer movement was found distracting by some (*"it kept moving around erratically and didn't seem to have any pattern."*) and some participants felt like being "watched" (*"I think I prefer to do things on my own without feeling like I'm being 'watched' but it was right there moving the arrow around the whole time,"* and *"It felt like someone's spying."*)

Finally, the full Pointer Assistant interaction (*proactive chatbot with pointer*) was specifically perceived by some as "fun" as participants noted that *"I thought it was fun and efficient," "It was fun and engaging to have an AI chatbot helping to remember what to include and where to include them. I loved it,"* and *"I thought it was fun, humorous, and helpful."*

7 Discussions and Future Work

This paper presented Pointer Assistant, a novel interaction technique for LLM-based human-AI interaction featuring two key characteristics: 1) *pointer form*, where the AI companion is displayed as an extra mouse cursor on the screen that can move to point at relevant areas, and 2) *proactivity*, where the AI responds to user actions on the screen without the user having to directly initiate the chat conversation. This design aims to facilitate human-AI collaboration in complex on-screen tasks that could benefit from real-time feedback.

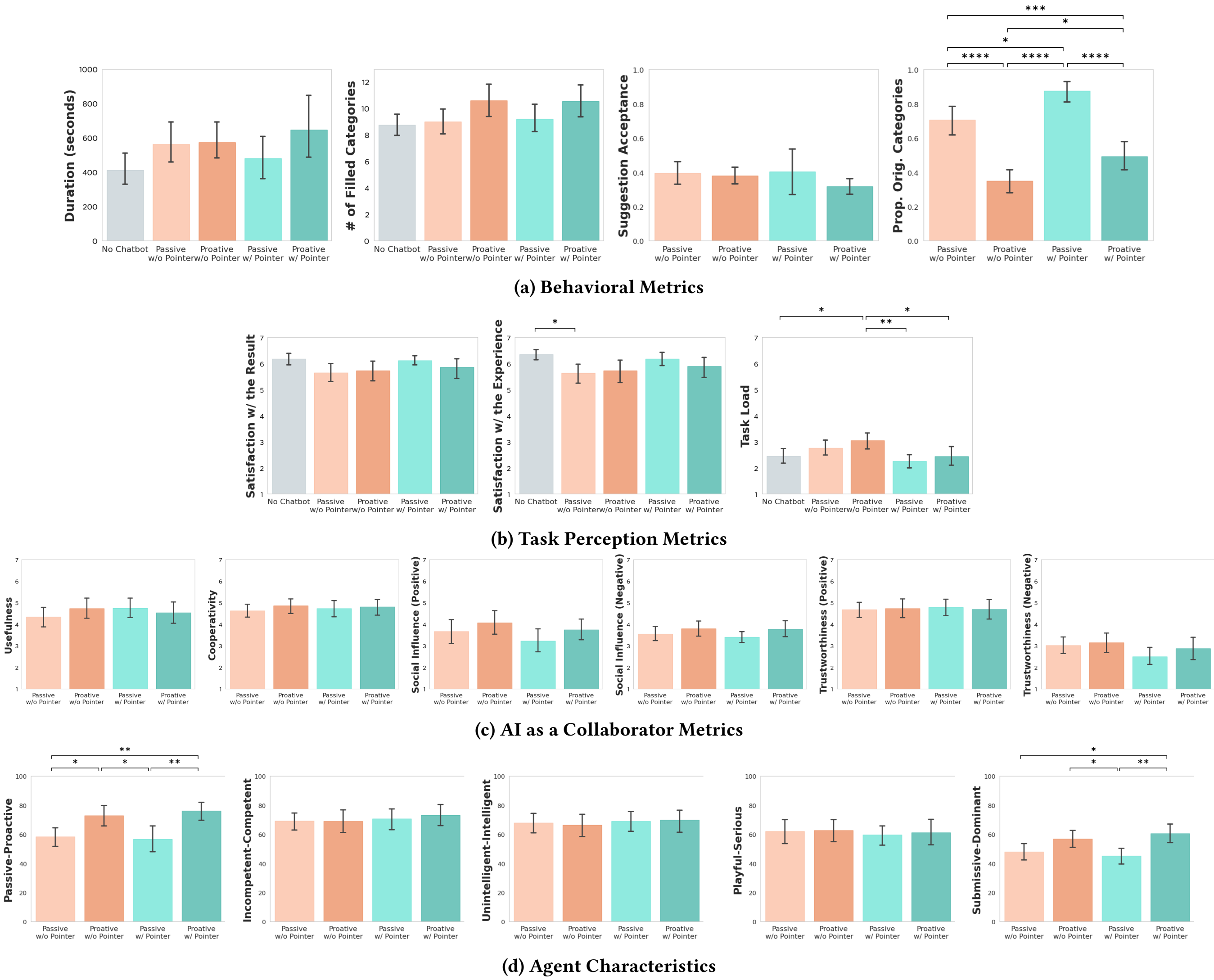


Figure 6: Measured metrics by group. Significant pairwise differences are denoted ($^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$). Differences between other pairs were not statistically significant ($p > 0.05$)

7.1 Experimental Results and Design Implications

The results from the exploratory study assessing the effectiveness of this interaction technique using financial budget planning as an example task revealed contributions that each key characteristic of Pointer Assistant, pointer form and proactivity, offers to the overall user experience and behavior.

The statistical analyses indicated the pointer form led to increased satisfaction with the experience with the AI chatbot and reduced perceived task load. This can be explained through the qualitative responses noting that the pointer chatbot felt "fun" and "engaging" to use, and how its spatial movements can visually guide users through each step of the experience, reducing unnecessary mental efforts. This may also be attributed to the hypothesized benefit of not having to context-switch between the task and interacting with the chatbot. However, the perception of the interaction being fun may also be influenced by the mere novelty of the interaction. Interestingly, the pointer form also increases the proportion of original categories compared to their non-pointer counterparts. This may be due to the transient nature of the chat bubble at the end of the cursor, making users rely less on the given suggestions when compared to a static chat history log that they can always refer back to. This might suggest an additional benefit of the ephemeral display of AI's recommendations through pointer form, as the AI's feedback is shown just long enough to spark ideas, but not long enough for the user to over-rely on the model's recommendations.

As for proactivity, first, the passive-proactive scale showed that both proactive chatbot conditions were perceived to be significantly more proactive as designed. Furthermore, this trait helped

Table 4: Regression coefficients (β) and 95% confidence interval on behavioral and task perception metrics with intervention conditions as predictors: Proactive intervention increases the number of filled categories and decreases the proportion of original categories while Pointer intervention increases the proportion of original categories and satisfaction with the experience while reducing perceived task load.

	Behavioral Metrics				Task Perception (1-7 Likert)		
Condition	Duration (seconds)	# of Filled Categories	Suggestion Acceptance (0-1)	Prop. Orig. Categories (0-1)	Satisfaction w/ the Result	Satisfaction w/ the Experience	Task Load
Pointer	-81.825 [$-280.851, 117.202$] ($p = 0.418$)	0.194 [$-1.414, 1.802$] ($p = 0.812$)	0.009 [$-0.105, 0.123$] ($p = 0.876$)	0.168^{**} [$0.060, 0.277$] ($p = 0.003$)	0.460 [$-0.012, 0.932$] ($p = 0.056$)	0.554^{*} [$0.019, 1.089$] ($p = 0.042$)	-0.511^{*} [$-0.941, -0.081$] ($p = 0.020$)
Proactive	12.458 [$-184.241, 209.156$] ($p = 0.901$)	1.614^{*} [$0.025, 3.203$] ($p = 0.047$)	-0.014 [$-0.110, 0.082$] ($p = 0.777$)	-0.358^{***} [$-0.465, -0.250$] ($p < 0.001$)	0.068 [$-0.398, 0.534$] ($p = 0.773$)	0.091 [$-0.438, 0.620$] ($p = 0.735$)	0.288 [$-0.137, 0.713$] ($p = 0.183$)
Pointer ∗ Proactive	155.553 [$-125.912, 437.019$] ($p = 0.277$)	-0.256 [$-2.530, 2.017$] ($p = 0.824$)	-0.073 [$-0.217, 0.070$] ($p = 0.315$)	-0.025 [$-0.179, 0.129$] ($p = 0.750$)	-0.330 [$-0.997, 0.337$] ($p = 0.330$)	-0.377 [$-1.133, 0.380$] ($p = 0.327$)	-0.097 [$-0.705, 0.511$] ($p = 0.752$)

*p<0.05, **p<0.01, ***p<0.001

Table 5: Regression coefficients (β) and 95% confidence interval on "AI as a Collaborator" metrics with intervention conditions as predictors: No significant effects from the interventions were observed.

Intervention Term	Usefulness	Cooperativity	Social Influence (Positive)	Social Influence (Negative)	Trustworthiness (Positive)	Trustworthiness (Negative)
Pointer	0.406 [$-0.276, 1.088$] ($p = 0.242$)	0.098 [$-0.399, 0.595$] ($p = 0.697$)	-0.444 [$-1.183, 0.296$] ($p = 0.238$)	-0.152 [$-0.645, 0.342$] ($p = 0.546$)	0.104 [$-0.496, 0.705$] ($p = 0.732$)	-0.522 [$-1.175, 0.131$] ($p = 0.116$)
Proactive	0.394 [$-0.280, 1.068$] ($p = 0.250$)	0.227 [$-0.264, 0.718$] ($p = 0.362$)	0.398 [$-0.333, 1.129$] ($p = 0.284$)	0.239 [$-0.249, 0.727$] ($p = 0.336$)	0.057 [$-0.537, 0.650$] ($p = 0.850$)	0.125 [$-0.520, 0.770$] ($p = 0.703$)
Pointer ∗ Proactive	-0.608 [$-1.572, 0.356$] ($p = 0.215$)	-0.152 [$-0.855, 0.551$] ($p = 0.670$)	0.126 [$-0.920, 1.172$] ($p = 0.812$)	0.130 [$-0.568, 0.829$] ($p = 0.713$)	-0.152 [$-1.001, 0.697$] ($p = 0.724$)	0.256 [$-0.668, 1.180$] ($p = 0.585$)

*p<0.05, **p<0.01, ***p<0.001

Table 6: Regression coefficients (β) and 95% confidence interval on "Agent Characteristics" metrics with intervention conditions as predictors: Proactive intervention is a significant positive predictor of the chatbot's perceived proactiveness and dominance.

Intervention Term	Passive (0) –Proactive (100)	Incompetent (0) –Competent (100)	Unintelligent (0) –Intelligent (100)	Playful (0) –Serious (100)	Submissive (0) –Dominant (100)
Pointer	-1.732 [$-12.354, 8.891$] ($p = 0.748$)	1.634 [$-8.596, 11.864$] ($p = 0.753$)	1.093 [$-9.301, 11.487$] ($p = 0.836$)	-2.511 [$-13.629, 8.607$] ($p = 0.656$)	-2.944 [$-11.480, 5.593$] ($p = 0.497$)
Proactive	14.318^{**} [$3.820, 24.817$] ($p = 0.008$)	-0.227 [$-10.338, 9.883$] ($p = 0.965$)	-1.364 [$-11.636, 8.909$] ($p = 0.794$)	0.682 [$-10.306, 11.670$] ($p = 0.903$)	8.864^{*} [$0.427, 17.300$] ($p = 0.040$)
Pointer ∗ Proactive	4.968 [$-10.055, 19.990$] ($p = 0.515$)	2.608 [$-11.859, 17.076$] ($p = 0.722$)	2.316 [$-12.383, 17.015$] ($p = 0.756$)	0.985 [$-14.739, 16.708$] ($p = 0.902$)	6.613 [$-5.460, 18.685$] ($p = 0.281$)

*p<0.05, **p<0.01, ***p<0.001

participants come up with more categories compared to their passive counterparts. As participants generally mentioned that these proactive chatbots were helpful in reminding them about budget categories they might have missed, we speculate that this design contributed to this increase through the higher number of suggestions given, which resulted from the fact that no user initiation is required due to the proactive setting. In contrast, proactive chatbots significantly reduced the proportion of original categories while also being perceived as more dominant. This may be because of the sheer number of suggestions and how frequently they are presented to the user, as voiced by the users in proactive conditions that the proactive assistants can feel pushy at times.

Another interesting finding was that the Pointer Assistant design and its deconstructed versions in each condition did not result in

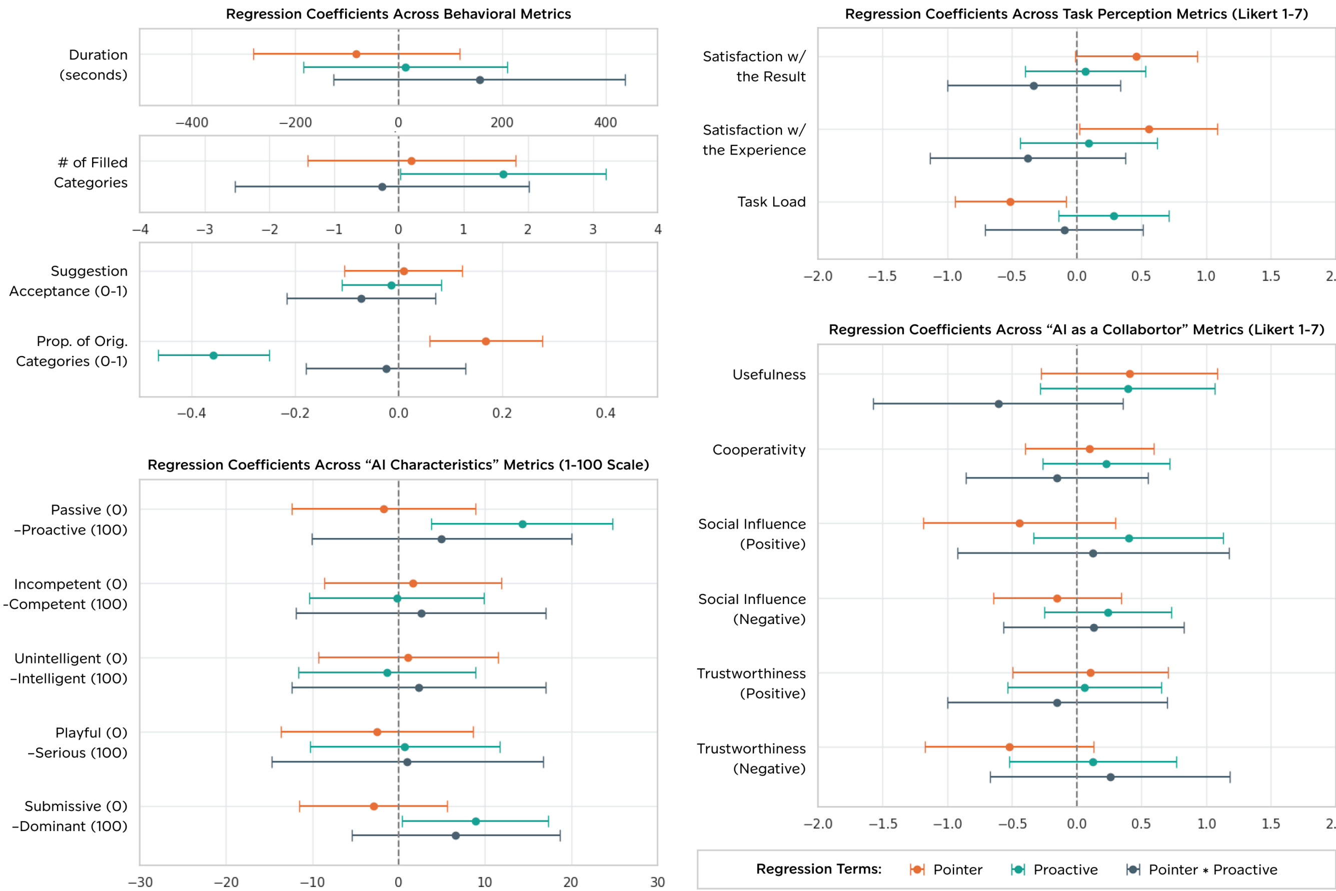


Figure 7: Coefficient plots of estimates for the regression terms (Proactive, Pointer) and their interaction (Proactive ∗ Pointer) across each group of measured metrics, with 95% confidence intervals.

significant differences in metrics reflecting perception towards the AI assistant in terms of usefulness, cooperativity, social influence, trustworthiness, perceived assistant's competence, intelligence, and seriousness as hypothesized. Future inspections could explore other characteristics to better understand the driving forces behind the other significant outcomes.

Overall, we have shown that the proposed Pointer Assistant interaction has achieved its goals of improving the popular design for human-AI interaction of passive chat log–based LLM chatbots by 1) reducing users' task load with visual guidance and less context-switching, and 2) improving satisfaction with the experience while enhancing users' productivity through the AI-generated real-time feedback. These attributes can be useful for on-screen spatial tasks beyond financial budgeting such as brainstorming, programming, writing, and design tasks, as exemplified in the previous sections. However, it is also important to note that the reduced proportion of original budget categories in this study may indicate the potential of this design to make users become over-reliant on the AI assistant's responses while the momentary display of AI's suggestions may be able to slightly help alleviate the negative effects. With the results from this study, interaction designers and researchers can employ parts of this interaction technique in their designs of AI-assisted applications as an alternative to a traditional passive user-AI chat log to achieve these effects in their designs.

In addition, it is also worth noting that these gains are mostly in comparison to the traditional chat log–based chatbot user interface design. In terms of the benefits of AI assistants over the non-AI interface, our results were not able to clearly show significant improvements of using Pointer Assistant over the *no chatbot* control condition in most measures. On top of that, though not apparent in the full Pointer Assistant mode, we even found a significant increase in task load in the *proactive chatbot without pointer* condition compared to the one without any AI assistants. While this may be due to the small effect size combined with the limited sample size, it emphasizes the importance of the designer's role in verifying the actual benefits and effectiveness of adding AI and LLM-based assistants to any task.

7.2 System Limitations

While this paper has shown the benefits and other considerations of the Pointer Assistant interaction technique, there are system limitations that should be noted.

Firstly, there is still room for refinement in terms of user interaction. For example, the chatbot may sometimes repetitively give the same suggestions leading to user annoyance. This may be refined through better prompting techniques or adding logical checks such as filtering out responses on the same issue recently mentioned. The cursor movement can also be fine-tuned or changed to other ways of spatially displaying chatbot responses to make it less distracting. For example, the AI pointer might only appear when it is conversing with the user. Proactivity and the timing of the response can also be improved to reduce the chatbot's unwanted dominance and pushiness. For instance, the model might respond to users' actions more selectively; not responding to trivial user actions.

Secondly, our current implementation of Pointer Assistant was based on structured or tabular user interfaces. This is thanks to how the user data presented on the interface can easily be converted into JSON format that could be sent to an AI model and mapped back to the location on the screen for pointer movement. While it may take some mild effort to apply this technique to other structured interfaces, such as spreadsheets, web forms, or prose writing (as described earlier), it can be more challenging to map AI's responses to relevant screen coordinates in other types of interfaces, especially a more visual ones such as a drawing canvas. This is because current AI models cannot reliably specify a coordinate or bounding box of interest when given an image. Future work could explore implementation techniques that enable Pointer Assistant on other types of user interfaces to unlock a wider range of use cases.

Finally, another notable limitation is that the quality of the chatbot's responses heavily depended on the LLM's quality thanks to its LLM-based nature. Whether the model's comments will be useful relies extensively on comprehensive prompts that inform both the style and substance of the responses. This may be addressed through iterative testing and careful prompt crafting to include sufficiently detailed instructions for the chatbot as well as specific domain knowledge of the task. However, it is still possible that the chatbot may respond with inaccurate information due to so-called "hallucinations" of LLMs. This risk could be reduced through prompting techniques or other techniques such as data Retrieval-Augmented Generation (RAG) [18], or even future advances in artificial intelligence. While we have not found instances of major inaccuracies in our study—possibly because our pilot use case was relatively simple, this is a notable consideration in future studies.

7.3 Future Work

Future research could build upon this work in multiple directions. One such direction is to address the system limitations mentioned above to improve user experience, widen potential use cases, and improve the model's quality of responses.

Another possible direction involves conducting additional studies to explore other aspects of this design and fill the gaps in this study. Limitations of this study that can be addressed in future works include the limited sample size (<50 participants per condition) and narrow task domain (financial planning). For example, larger sample sizes may lead to a clearer statistical significance in some metrics, and applying Pointer Assistant to tasks other than budgeting, e.g., planning, programming, and creative tasks, may lead to different results that help us better understand the strengths and weaknesses of this design. Moreover, long-term studies can help highlight important characteristics or other possible adverse effects of this interaction technique.

Lastly, future research could also explore pushing this interaction technique even further, e.g., by not only having the model respond textually but also taking actions, with or without users' direct prompts. This could enable new possibilities for a more seamless human-AI collaboration. For instance, while working on a spreadsheet, the user may ask the on-screen cursor assistant to edit certain parts and collaboratively check each other's work, potentially resulting in boosted productivity and quality of the outcomes.

By exploring new human-AI interaction techniques like Pointer Assistant, we are taking steps towards reimagining how humans and artificial intelligence can collaborate more intuitively. Our work demonstrates that interaction design can transform AI from a static tool to a dynamic, context-aware partner that actively supports its users with their tasks through guidance, stimulation for ideation, and real-time feedback. As this field evolves, we move closer to creating truly helpful human-AI interfaces that enhance both productivity and quality of the task's outcomes while maintaining a satisfactory user experience.

8 Conclusion

We presented Pointer Assistant, a novel interaction technique for LLM-based human-AI interaction with cursor form and proactivity as its key characteristics. This design aimed at the ability for LLM models to proactively provide users with real-time spatial guidance, ideation stimuli, and feedback without causing too much mental load due to the switching between their tasks and a separate chat screen on the side. This proposed technique can also be adapted to different tasks in multiple domains such as programming, design, and education. Our exploratory study with 220 participants involving an example task of financial budget planning revealed significant improvements compared to traditional chatbot interfaces in areas including user satisfaction with the process, reduced task load, and enhanced productivity—measured through a greater number of budget categories the user came up with. The proactive nature of the chatbot also made the assistant perceived as more proactive and dominant in the task. Qualitative results also indicated that the participants found Pointer Assistant useful in reminding them about budget categories they might have missed, beneficial in giving real-time feedback to their input, helpful in providing visual guidance through the user interface, but also potentially overly assertive due to its proactive nature. These results highlight the possibility of designing new interaction techniques for LLM-based AI models to improve certain characteristics that may be beneficial to the user. Further development of this interaction technique as well as additional studies examining this technique could offer richer insights into better innovative designs for human-AI interactions to enhance both the user experience and the task outcomes.

References

[1] Christiane Attig, Patricia Wollstadt, Tim Schrills, Thomas Franke, and Christiane B. Wiebel-Herboth. 2024. More than Task Performance: Developing New Criteria for Successful Human-AI Teaming Using the Cooperative Card Game Hanabi. In *Extended Abstracts of the 2024 CHI Conference on Human Factors in Computing Systems (CHI EA '24)*. Association for Computing Machinery, New York, NY, USA, Article 245, 11 pages. https://doi.org/10.1145/3613905.3650853

[2] Renaud Blanch and Michaël Ortega. 2009. Rake cursor: improving pointing performance with concurrent input channels. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) *(CHI '09)*. Association for Computing Machinery, New York, NY, USA, 1415–1418. https://doi.org/10.1145/1518701.1518914

[3] Xiang Cao, Nicolas Villar, and Shahram Izadi. 2010. Comparing user performance with single-finger, whole-hand, and hybrid pointing devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '10)*. Association for Computing Machinery, New York, NY, USA, 1643–1646. https://doi.org/10.1145/1753326.1753571

[4] Nazli Cila. 2022. Designing Human-Agent Collaborations: Commitment, responsiveness, and support. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 420, 18 pages. https://doi.org/10.1145/3491102.3517500

[5] Izabel Cvetkovic, Valeria Rosenberg, and Eva Bittner. 2023. Conversational Agent as a Black Hat: Can Criticizing Improve Idea Generation?. In *Proceedings of the 56th Hawaii International Conference on System Sciences*. 187–196. https://doi.org/10.24251/HICSS.2023.023

[6] Valdemar Danry, Pat Pataranutaporn, Yaoli Mao, and Pattie Maes. 2023. Don't Just Tell Me, Ask Me: AI Systems that Intelligently Frame Explanations as Questions Improve Human Logical Discernment Accuracy over Causal AI explanations. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 352, 13 pages. https://doi.org/10.1145/3544548.3580672

[7] Fred D. Davis. 1989. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Q.* 13 (1989), 319–340. https://api.semanticscholar.org/CorpusID:12476939

[8] Graham Dove, Ayanna Seals, and Oded Nov. 2020. Socially-Informed Sorting for Guiding Personal Finance Choices. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '20)*. Association for Computing Machinery, New York, NY, USA, 1–9. https://doi.org/10.1145/3334480.3382898

[9] Upol Ehsan, Philipp Wintersberger, Q. Vera Liao, Elizabeth Anne Watkins, Carina Manger, Hal Daumé III, Andreas Riener, and Mark O Riedl. 2022. Human-Centered Explainable AI (HCXAI): Beyond Opening the Black-Box of AI. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI EA '22)*. Association for Computing Machinery, New York, NY, USA, Article 109, 7 pages. https://doi.org/10.1145/3491101.3503727

[10] Alex Endert, Patrick Fiaux, Haeyong Chung, Michael Stewart, Christopher Andrews, and Chris North. 2011. ChairMouse: leveraging natural chair rotation for cursor navigation on large, high-resolution displays. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI EA '11)*. Association for Computing Machinery, New York, NY, USA, 571–580. https://doi.org/10.1145/1979742.1979628

[11] Figma. 2021. Send messages with cursor chat — help.figma.com. https://help.figma.com/hc/en-us/articles/1500004414842-Send-messages-with-cursor-chat. [Accessed 07-09-2024].

[12] Susumu Harada, Jacob O. Wobbrock, Jonathan Malkin, Jeff A. Bilmes, and James A. Landay. 2009. Longitudinal study of people learning to use continuous voice-based cursor control. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) *(CHI '09)*. Association for Computing Machinery, New York, NY, USA, 347–356. https://doi.org/10.1145/1518701.1518757

[13] Rune Møberg Jacobsen, Lukas Bjørn Leer Bysted, Patrick Skov Johansen, Eleftherios Papachristos, and Mikael B. Skov. 2020. Perceived and Measured Task Effectiveness in Human-AI Collaboration. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '20)*. Association for Computing Machinery, New York, NY, USA, 1–9. https://doi.org/10.1145/3334480.3383104

[14] Elizabeth M. Kanny and Denis Anson. 1991. A pilot study comparing mouse and mouse-emulating interface devices for graphic input. *Assistive Technology* 3, 2 (1991), 50–58. https://doi.org/10.1080/10400435.1991.10132182

[15] Hanseob Kim, Bin Han, Jieun Kim, Muhammad Firdaus Syawaludin Lubis, Gerard Jounghyun Kim, and Jae-In Hwang. 2024. Engaged and Affective Virtual Agents: Their Impact on Social Presence, Trustworthiness, and Decision-Making in the Group Discussion. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 149, 17 pages. https://doi.org/10.1145/3613904.3642917

[16] Manu Kumar, Andreas Paepcke, and Terry Winograd. 2007. EyePoint: practical pointing and selection using gaze and keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '07)*. Association for Computing Machinery, New York, NY, USA, 421–430. https://doi.org/10.1145/1240624.1240692

[17] F. J. Lerch, M. M. Mantei, and J. R. Olson. 1989. Skilled financial planning: the cost of translating ideas into action. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '89)*. Association for Computing Machinery, New York, NY, USA, 121–126. https://doi.org/10.1145/67449.67474

[18] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) *(NIPS '20)*. Curran Associates Inc., Red Hook, NY, USA, Article 793, 16 pages.

[19] I. Scott MacKenzie, Abigail Sellen, and William A. S. Buxton. 1991. A comparison of input devices in element pointing and dragging tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New Orleans, Louisiana, USA). Association for Computing Machinery, New York, NY, USA, 161–166. https://doi.org/10.1145/108844.108868

[20] Miro. 2019. Miro | The Innovation Workspace — miro.com. https://miro.com/. [Accessed 12-09-2024].

[21] M. J. Muller. 1988. Multifunctional cursor for direct manipulation user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Washington, D.C., USA) *(CHI '88)*. Association for Computing Machinery, New York, NY, USA, 89–94. https://doi.org/10.1145/57167.57181

[22] Anthropic PBC. 2024. Introducing computer use, a new Claude 3.5 Sonnet, and Claude 3.5 Haiku — anthropic.com. https://www.anthropic.com/news/3-5-models-and-computer-use. [Accessed 08-12-2024].

[23] Barry A. Po, Brian D. Fisher, and Kellogg S. Booth. 2005. Comparing cursor orientations for mouse, pointer, and pen interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Portland, Oregon, USA) *(CHI '05)*. Association for Computing Machinery, New York, NY, USA, 291–300. https://doi.org/10.1145/1054972.1055013

[24] Francis K. H. Quek and Michael C. Petro. 1993. Human-machine perceptual cooperation. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems* (Amsterdam, The Netherlands) *(CHI '93)*. Association for Computing Machinery, New York, NY, USA, 123–130. https://doi.org/10.1145/169059.169116

[25] Louis B. Rosenberg, Jonathan L. Beamer, Adam C. Braun, and Dean C. Chang. 1998. Interface device and method for providing enhanced cursor control with force feedback. https://patents.google.com/patent/US6252579

[26] Advait Sarkar. 2024. AI Should Challenge, Not Obey. *Commun. ACM* 67, 10 (Sept. 2024), 18–21. https://doi.org/10.1145/3649404

[27] Kay Schroeder, Batoul Ajdadilish, Alexander P. Henkel, and André Calero Valdez. 2020. Evaluation of a Financial Portfolio Visualization using Computer Displays and Mixed Reality Devices with Domain Experts. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–9. https://doi.org/10.1145/3313831.3376556

[28] Lauren Schwahn. 2022. Zero-Based Budgeting: Spend Every Penny but Meet Your Financial Goals - NerdWallet — nerdwallet.com. https://www.nerdwallet.com/article/finance/zero-based-budgeting-explained. [Accessed 12-09-2024].

[29] Lisa Strausfeld. 1995. Financial Viewpoints: using point-of-view to enable understanding of information. In *Conference Companion on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '95)*. Association for Computing Machinery, New York, NY, USA, 208–209. https://doi.org/10.1145/223355.223524

[30] Hariharan Subramonyam, Jane Im, Colleen Seifert, and Eytan Adar. 2022. Solving Separation-of-Concerns Problems in Collaborative Design of Human-AI Systems through Leaky Abstractions. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 481, 21 pages. https://doi.org/10.1145/3491102.3517537

[31] Carlos Toxtli, Andrés Monroy-Hernández, and Justin Cranshaw. 2018. Understanding Chatbot-mediated Task Management. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3173574.3173632

[32] Lindsey Vanderlyn, Gianna Weber, Michael Neumann, Dirk Väth, Sarina Meyer, and Ngoc Thang Vu. 2021. "It seemed like an annoying woman": On the Perception and Ethical Considerations of Affective Language in Text-Based Conversational Agents. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, Arianna Bisazza and Omri Abend (Eds.). Association for Computational Linguistics, Online, 44–57. https://doi.org/10.18653/v1/2021.conll-1.4

[33] Dakuo Wang, Elizabeth Churchill, Pattie Maes, Xiangmin Fan, Ben Shneiderman, Yuanchun Shi, and Qianying Wang. 2020. From Human-Human Collaboration to Human-AI Collaboration: Designing AI Systems That Can Work Together with People. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '20)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3334480.3381069

[34] Qian Zhou, George Fitzmaurice, and Fraser Anderson. 2022. In-Depth Mouse: Integrating Desktop Mouse into Virtual Reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 354, 17 pages. https://doi.org/10.1145/3491102.3501884