

Beyond ChatBots: EXPLORELLM for Structured Thoughts and Personalized Model Responses

XIAO MA*, Google, USA
 SWAROOP MISHRA†, Google Deepmind, USA
 ARIEL LIU, Google, USA
 SOPHIE SU, Google, USA
 JILIN CHEN, Google, USA
 CHINMAY KULKARNI, Emory University, USA
 HENG-TZE CHENG, Google Deepmind, USA
 QUOC LE, Google Deepmind, USA
 ED CHI, Google Deepmind, USA

The interface shows a user input: "i want to plan a trip to tokyo". Below this, a search bar contains the text "Ask a question to start the brainstorm". The main area displays eight sub-tasks in pink boxes, arranged in two rows of four:

- Identify key dates and duration for the Tokyo trip
- Research on art museums and galleries in Tokyo
- Look for popular ramen places in Tokyo
- Look into available and affordable accommodations in Tokyo
- Look into flight options and prices to Tokyo
- Identify necessary documents for travel to Japan
- Plan daily itineraries involving art galleries and ramen shops
- Investigate public transportation options in Tokyo

Below the sub-tasks, a text input field contains "I like art and ramen". A green button labeled "Re-generate ideas" is positioned below the input field. At the bottom, a blue button labeled "Summarize" is visible.

Fig. 1. EXPLORELLM introduces a new interaction pattern with large language models (LLMs) by automatically decomposing complex tasks into sub-tasks, and allowing users greater task control and personalization.

Large language model (LLM) powered chatbots are primarily text-based today, and impose a large interactional cognitive load, especially for exploratory or sensemaking tasks such as planning a trip or learning about a new city. Because the interaction is textual, users have little scaffolding in the way of structure, informational “scent”, or ability to specify high-level preferences or goals. We introduce EXPLORELLM that allows users to structure thoughts, help explore different options, navigate through the choices and recommendations, and to more easily steer models to generate more personalized responses. We conduct a user study and show that users find it helpful to use EXPLORELLM for exploratory or planning tasks, because it provides a useful

* Author list is by descending order of contribution.

† Developed initial idea of prompt decomposition and tree-based design and interaction.

Authors’ addresses: Xiao Ma, xmaa@google.com, Google, USA; Swaroop Mishra, swaroopmishra@google.com, Google Deepmind, USA; Ariel Liu, arielliu@google.com, Google, USA; Sophie Su, sophiesu@google.com, Google, USA; Jilin Chen, jilinc@google.com, Google, USA; Chinmay Kulkarni, chinmay.kulkarni@emory.edu, Emory University, USA; Heng-Tze Cheng, hengtze@google.com, Google Deepmind, USA; Quoc Le, qvl@google.com, Google Deepmind, USA; Ed Chi, edchi@google.com, Google Deepmind, USA.

schema-like structure to the task, and guides users in planning. The study also suggests that users can more easily personalize responses with high-level preferences with EXPLORELLM. Together, EXPLORELLM points to a future where users interact with LLMs beyond the form of chatbots, and instead designed to support complex user tasks with a tighter integration between natural language and graphical user interfaces.

CCS Concepts: • **Human-centered computing** → **Graphical user interfaces; Natural language interfaces; Collaborative interaction**; *Empirical studies in interaction design*.

Additional Key Words and Phrases: Chatbots, Artificial Intelligence, Large Language Models, Natural Language Interfaces, Task Decomposition, Graphical User Interfaces, Interaction, Schema, Prompting, Learning from Instruction.

1 INTRODUCTION

Large language model (LLMs) powered chatbots have dramatically improved the user adoption of AI systems but have limited interaction patterns that are linear and text-heavy. Users can only carry out a single-stream conversation with existing chatbots such as Google Bard or OpenAI ChatGPT. These chatbots, for the large part, respond with text¹. Such textual responses, which are often verbose, impose a significant cognitive load on users to understand and act upon, especially for complex tasks. Often, users have to engage in multi-turn conversations, both to probe what the chatbots can understand, and to communicate their intent and preferences. Similarly, because users can only respond with text, conversational repair [1] is effortful.

While there have been significant advances in prompt-based methods that unlock the reasoning and planning abilities of LLMs [29, 47, 50, 53, 56], the interaction pattern between users and LLM-based assistants has largely remained the same. This remains the case despite increasing evidence that users struggle to communicate complex tasks to assistants [4, 16], much like conversational assistants that preceded them [57]. Just as non-AI-experts use ad-hoc repair strategies to improve prompts for LLMs [54], non-expert users similarly use ad-hoc tactics like adding details to their request, pointing out assistant errors in how the request was interpreted, or simply giving up on their original task and deviating to a related, simpler task [16]. These tactics sometimes act as a “band-aid”, but still leave a majority of users unsatisfied [16]. More importantly, they prevent users from fully leveraging the potential of AI assistance to complete their tasks.

In this work, we introduce a new interaction pattern between users and LLM-powered assistants, by combining a prompt-based *task decomposition* method with a new schema-like graphical user interface (UI). The new system, EXPLORELLM, decomposes tasks into sub-tasks *automatically* using a prompt-based decomposition method. Building on the LLM reasoning literature [47, 50, 53, 56], we custom-designed prompts to generate related and easier-to-solve sub-tasks related to the original user query. Then, inspired by theories of schema in cognitive science [26] and distributed sensemaking [8] in human-computer interaction, we render the generated sub-tasks for the users in a structured and interactive UI. As a concrete example, for a complex task such as “I want to plan a trip to Tokyo”, EXPLORELLM will organize the query into sub-tasks such as deciding on dates and duration, making hotel and flight arrangements, checking travel documents, etc. Such organizational structures, or schema, allow people to not only learn what aspects are important in the given task, but also act as a cue to express their own preferences. Further, we design a dedicated user preference solicitation UI and a recommender-like user interaction within sub-tasks, aiming at improving personalization and the ability of users to easily steer model responses.

We conducted a user study with eight participants where we asked users to compare ChatGPT and EXPLORELLM on a planning task. Our user study results show that EXPLORELLM is helpful in providing structured task breakdown which helps users to think and structure their thoughts.

¹As of Nov 2023, the output of chatbots are becoming increasingly multimedia, but single-stream and text-heavy nonetheless.

Users find the structured guidance that EXPLORELLM provides useful for planning. Users mention that EXPLORELLM is easier to personalize with their own preferences, in contrast to text-based chatbots.

The rest of the manuscript is organized as follows. We first give an overview of the EXPLORELLM design, providing details on the key components and implementation details. Through a qualitative user evaluation, we describe how it helps users complete complex tasks. In later discussion, we outline future work on integrating tool use and further opportunities for automation. Finally, we discuss the limitations of EXPLORELLM, some of which stem from foundational limitations of LLMs (such as hallucination), and others from the limited functionality that we developed. We plan to open-source EXPLORELLM following the publication of this manuscript.

2 BACKGROUND

EXPLORELLM builds on recent work in LLM reasoning, theories from human cognition, and prior work on natural language and graphical user interfaces in human-computer interaction (HCI).

2.1 Prompting elicits reasoning and planning in LLMs

In-context learning [2] and its evolution via various prompting methods have unlocked the reasoning and planning abilities of LLMs. Once instruction-tuned [30, 35, 49], LLMs can follow specific instructions from end-users in natural language. Leveraging the instruction following abilities of LLMs, researchers show that carefully designed prompts can improve LLM performance across a variety of reasoning and planning tasks, through intermediate representation in the form of thoughts [32, 50], decomposition [15, 37, 39, 56], search-decomposition mix [41, 53], structure [5, 9, 28], abstraction [55] and optimization [52]. Prompting based solutions have significantly improved model capabilities across a diverse range of tasks such as program synthesis [19], dialogue [11], biomedical applications [36], style transfer [40], multilingual [43], moral reasoning [22] and a large spectrum of natural language understanding tasks [24, 48]. However, most of these advancements are targeted at improving LLMs’ performance on some benchmark tasks, rather than benefiting end users (non-expert users in particular [54]). In this work, we explore the possibility of leveraging prompt-based methods to better support end users in their complex exploratory or planning tasks.

2.2 Schemata support thinking and problem solving

Interestingly, some methods for eliciting the reasoning ability in LLMs have roots in psychology and cognitive science – particularly the concept of *schema*. A schema is a framework, outline, or plan for solving a problem [26]. Prior work shows that schema is an effective tool in supporting human problem solving [23, 38]. One way to create schemata is through task decomposition. The intuition of breaking tasks down also aligns well with the distributed sensemaking work, which indicates that in solving complex problems, it is useful for people to have a starting problem structure, that they can later customize to their own goals [8]. Further, although originally developed by cognitive scientists to describe human learning [46], the concept of schema has inspired subsequent frameworks of machine intelligence, such as work by Minsky on *frame* and *frame-systems* [27]. When faced with a new situation, humans select from memory a structure called a *frame*, a data-structure representing a stereotyped situation. Collections of related frames are linked together into *frame-systems* that can be manipulated while allowing information coordination [27]. In this work, we leverage LLMs’ reasoning ability to assist humans in creating schemata for problem solving using prompt-based task decomposition automatically. In addition, we draw inspirations from frames and frame-systems when designing system interfaces. Users can interact with each

sub-task as connected components via a UI for further customization and the EXPLORELLM system keeps track of user contexts and preferences for coordinated decision making.

2.3 Why natural language alone is not enough

Natural language user interfaces (NLUIs) and graphical user interfaces (GUIs) are two major ways for humans to interact with machines. As early as 1972, Winograd developed SHRDLU, a natural-language interface that manipulates blocks in a virtual “blocks world” [51]. The effectiveness of NLUIs are limited by the capability of underlying AI systems. The invention of GUIs in the 1970s was largely a response to the lack of the natural language understanding and generation abilities of machines. GUIs played a major role in the wide adoption of personal computing [13]. With the recent advancements in LLMs, NLUIs received renewed attention in the form of chatbots.

At the same time, there is compelling evidence that natural language interfaces alone are not enough: decades of work in cognitive science suggests that thinking is intimately tied to *doing*, not just speaking. Put differently, thinking is a process that is not limited to what happens in our brain, but instead it is “distributed” throughout our environment [12]. For example, people find it much easier to move lettered tiles into various arrangements for playing Scrabble [25] (rather than merely talk through alternatives), and professionals frequently draw rough diagrams to “help them think” [23].

While natural language is flexible, it limits users mostly to single-stream and text-heavy interaction patterns. GUIs have unique advantages that are more compatible with human cognition and sensemaking. Consistent with the schemata concept, graphical user interfaces are particularly helpful in adding structure to a task, and allowing users to notice aspects of the task that matter the most [10, 45]. For instance, a table comparing alternatives across the most important dimensions helps programmers choose between competing technical approaches [21], and a “mind map” developed by other users can help users learn important aspects of a complex problem [8]. Over time, if interfaces are well-structured and predictable, users may develop tool-specific expertise in using them that extends beyond their conceptual task understanding [18]. For example, photo-editing experts develop expertise with specific tools like Adobe Photoshop that goes beyond a conceptual understanding of editing photos. Unfortunately, current AI chatbot interfaces do not take advantage of GUIs, and instead generate responses to each query linearly. As a result, users struggle to develop a strong mental model of such interactions, which are especially important for complex tasks.

In this work, we introduce EXPLORELLM to automatically induce a structure that highlights salient dimensions for exploratory tasks, by combining the best of natural language and graphical user interfaces. Some recent work has started exploring the better design of graphical interfaces for LLMs [14, 44]. For example, Sensecape is an interactive system built with LLMs to support flexible information foraging and sensemaking [44]. Graphologue converts the text-only response from LLMs to interactive diagrams to facilitate information-seeking and question-answering tasks [14]. EXPLORELLM builds on these initial explorations to create predictable, structured interfaces for complex exploratory tasks, focusing on task decomposition. Together, we demonstrate a promising direction of “hybrid” user interfaces, where a tighter integration is drawn between natural language and graphical user interfaces: language allows for expression of complex and abstract goals and preferences, and graphical representations allow for more structured understanding and exploration [17, 18].

3 METHODS

In this section, we first outline the key design components of the EXPLORELLM system and implementation details. Then we describe the setup of the user study.

3.1 EXPLORELLM Overall Design

We first provide an overview of the system, and then delve into details and the design rationales of each component. In this work, we design a standalone prototype for simplicity. However, this mode of interaction can potentially be invoked from the current chatbot interfaces with magic commands such as “/explore”.

3.1.1 System Overview.

- (1) **Node:** The EXPLORELLM system has an underlying tree-like data structure. Unlike traditional chatbots, we create an abstraction of a node that can be nested. A node is a unit of interaction and can represent different forms of interactions (e.g., multi-turn natural language chats or UI interfaces). By default, a new node is created when the user starts interacting with the system. When needed, the system automatically creates children nodes for users to explore through task decomposition. All nodes form a tree-like hierarchical structure that holds all the context of the user’s exploration journey. For this work, we limit the nodes to depth = 2 for simplicity (root user query with one layer of children nodes). In the future this can be extended to more layers or graphs.
- (2) **Personal Preferences:** At any given node, the users can provide free-form context that the system should be aware of for better personalization. The personal preference context is shared globally across all nodes.
- (3) **Options:** In each node, the system will take personal preferences into consideration and dispatch a backend call to present some options for the user to choose from. Users can interact with different options via a checkbox UI to indicate their preferences.
- (4) **Summarize:** After sufficient exploration, users may want to tie everything back together and get a summary of their journey so far. Therefore, the system has a “summarize” function that is available on each page. Users can click on the button to exit to the root node and get a text summary of their entire interaction across the system.

3.1.2 Node. One of the most important design goals of the EXPLORELLM system is to better support complex and under-specified tasks that require exploration. We address the challenge that complex tasks require high cognitive load by creating a tree-like abstraction. Through reasoning literature, especially on decompositions [15, 37, 56], we know that LLMs are capable of decomposing a complex problem into a list of easier subproblems. In this work, we leverage task decomposition abilities of LLMs for *users’ benefit*, rather than as a method to improve LLM’s task accuracy. Users can type into a generic query box and the system will first create a root node for this query, and then automatically calls the LLM task decomposition endpoint to create a list of easier sub-tasks.

We use the prompt in Figure 2 for automatic task decomposition. The original user query and any additional personalization cues are passed to the LLM through this prompt (Figure 2). The output of the prompt is a list of sub-tasks that, together with the original task, forms a hierarchical tree-like structure. We display each sub-task to the users as “cards” that they can interact with (see Figure 3), mimicking a schema-like structure. When users hover over one of the card, a “see more” button appears. Like a discussion thread in online forums, users now can focus on each task separately while the system keeps track of the logical structure, thus reducing user’s mental load.

3.1.3 Personal Preferences. The second design goal of the system is to support better personalization. Complex tasks usually have important personal contexts or constraints that are important to the user, and are incredibly overwhelming to elicit in one go. LLMs are trained to be “general purpose”, which dictates that the responses are often tailored to an “average user”. This “regression toward the mean” causes the LLMs response to be generic and not personalized. While it is possible for LLMs to offer more personalized responses once the users clearly specifies their preferences,

```

I want to accomplish the main goal of: {text}
To better assist me, please break down the problem into sub-problems.
Each sub-problem should help me to solve the original problem.
Make it so that each sub-problem is not trivial and can be helpful.
Take my context and personalization cues to personalize the sub-problems.
Make sure each sub-problem is concise and less than 15 words.

Personalization Cue: {selected_options}
My Context: {user_context}

Output format (make sure only output a valid JSON object that can be parsed with javascript function JSON.
parse).
Do not include any `` or json'.
{{
  "sub_problems": A list of strings (max 8), each as a valid sub-query
}}

Output:

```

Fig. 2. Prompt used in EXPLORELLM to break a complex task down into structured sub-tasks.

The screenshot displays the EXPLORELLM interface. At the top, there is a search bar with the placeholder text "Ask a question to start the brainstorm". Below this, the user's query "I want to plan a trip to Tokyo" is entered. The system has generated eight sub-tasks, each represented by a pink card:

- Identify key dates and duration for the Tokyo trip
- Research on art museums and galleries in Tokyo
- Look for popular ramen places in Tokyo
- Look into available and affordable accommodations in Tokyo
- Look into flight options and prices to Tokyo
- Identify necessary documents for travel to Japan
- Plan daily itineraries involving art galleries and ramen shops
- Investigate public transportation options in Tokyo

Below the cards, there is a section for personalization with the prompt "Tell us about yourself and your preferences to get better recommendations." and the user's input "I like art and ramen". A green button labeled "Re-generate ideas" is positioned below the input. At the bottom of the interface, there is a light blue button labeled "Summarize".

Fig. 3. User starts interacting with the EXPLORELLM system by typing a query. The system automatically breaks down the original user query into sub-tasks using a custom prompt, and then create nodes that represent each sub-task. Users can see the sub-tasks rendered as “cards”, and interact with each one. A dedicated UI prompts the users to specify personal contexts and preferences for personalization. Once user is satisfied with the exploration, they can click the summarize button at the bottom and EXPLORELLM generates a summary of the user journey.

prior work in recommender systems show that users themselves often are unaware of their needs and often struggle to express them, especially in the beginning of a planning process [31].

To more prominently elicit user preferences, we include a dedicated UI that is always available for users to update their preferences. We prompt the user to “tell us about yourself and your preferences to get better recommendations”. Importantly, the personalized context field is always available regardless which node the user is exploring, so that they can update the context if the task at hand reminds them of some preferences. The personalized context is passed in the prompt to the LLMs in all subsequent backend calls for better personalization.

3.1.4 Options. Another design solution for reducing mental load while increasing personalization is providing users options to choose from. Again, we draw inspirations from recommender systems. Prior work notes that the cognitive load for users to provide accurate preferences and ratings for items is much greater than providing implicit feedback (e.g., selecting an option they prefer) [33]. We leverage this insight and construct a LLM options generation endpoint with the prompt in Figure 4.

```
User: {text}

== Instructions ==
The user wants to: {context}
Here is one of the sub-query to help answer the main query.
Go into details to help me with the sub-query.
Show me some options to personalize and choose from.
Be concrete and make sure the options are valid choices to finish the task in sub-query.

Personalization Cue: {selected_options}
My Context: {user_context}
When coming up with options, make sure they are diverse and representative of multiple demographics, cultures
, and view points.

Output format (make sure only output a valid JSON object):
Do not include any '```' or 'json'.
{{
  "recommended": Your recommendation. ,
  "options": A list of options (at least 5) for me to choose from. Each option is a single string. Provide
             helpful details. Don't include numbers or bullet points.
}}

== End of Instruction ==

User: {text}
Output:
```

Fig. 4. Prompt used in EXPLORELLM to provide options for users to choose from.

We included the request for “diverse and representative” inspirationally to make sure the options generated are inclusive [20]. However, due to the small sample size of our user studies, we did not formally evaluate whether such a prompt is effective in increasing the options diversity or swaying user preferences. We discuss this in more detail in limitations and future work.

Once the users click on a node to “see more”, the system redirects them to a whole page screen to focus on the sub-task at hand. In the backend, the system dispatches the call to the LLM options generation endpoint. Once the results come back, we display the generated options with a checkbox UI (see Figure 5). The checkbox UI is designed to make it as easy as possible for users to provide implicit feedback through selection for better recommendations. The system keeps track of user selections, and any subsequent prompts to the backend will include all user selections for better personalization. The design intends to resemble a more passive “browsing” experience to minimize the mental load for users.

3.1.5 Summarize. After sufficient exploration in each sub-task, users may want to come back to the main task and get a summary of their journey. Therefore, we include an UI for summarization. Once the user clicks on “summarize”, the system gathers all user interaction signals and passes it to the LLM summarization endpoint with the prompt in Figure 6. The system redirects the user back to the root node and produces a textual summary of the main task.

3.2 Implementation

We implemented the above design with Next.js 13 for the front-end, and OpenAI GPT-4 API and FastAPI as backend. It is important to note that the interaction design is agnostic to the LLM API of

Look for popular ramen places in Tokyo

Select the ones you find helpful to get a summary later.

Recommended

☒ Based on your interest in art and ramen, I recommend visiting Afuri Ramen in Ebisu. The place offers a unique yuzu-infused Ramen, and is nearby the Tokyo Metropolitan Museum of Photography. So, you can enjoy the art and then dine at Afuri.

Other Options:

☐ Afuri Ramen, Ebisu: Known for Yuzu Ramen, nearby Tokyo Metropolitan Museum of Photography.

☐ Tsuta, Sugamo: Michelin-starred restaurant famous for Shoyu and Shio Ramen. In vicinity to Somei Yoshino Sakura trees, a beautiful work of nature's art.

☐ Fuunji, Shinjuku: Popular for Tsukemen, close to Shinjuku Gyoen Park, where you can find beautiful gardens.

☐ Ichiran Ramen, Shibuya: Renowned for Classic Tonkotsu Ramen, and not too far from the Shibuya Crossing Street Art.

☒ Nakiryu, Toshima: Another Michelin-starred ramen restaurant, known for Dan Dan Noodles. The Rikugi-en garden, which is considered a natural work of art, is nearby.

Fig. 5. EXPLORELLM generates personalized options for users to choose from.

```
User: {text}
Your Response:

Here is some information helpful to know about the user to personalize response.
Personalization: {selected_options}
Context: {user_context}

Answer the original user query. When helpful, personalize the response.
```

Fig. 6. Prompt used in EXPLORELLM to generate personalized summary across different sub-tasks.

choice, as long as the LLM has sufficient instruction following and reasoning capabilities. User data is stored in a secured MongoDB database and only selected authors have access to the database. The services are hosted on Google Cloud Run and Vercel. We plan to open-source the implementation of EXPLORELLM following the publication of this manuscript.

3.3 User Studies

We conducted a qualitative study to evaluate EXPLORELLM. We used convenience sampling for recruiting participants. There was no restriction on age, gender, and the language was limited to English. We asked a few screening questions about prior experiences and attitudes towards AI chatbots, such as Bard by Google, ChatGPT by OpenAI and Claude by Anthropic.

Then, participants were instructed to do a two-part unmoderated user study. We asked the participants to record themselves during the study, while presenting their computer screens and thinking aloud. Participants were informed of their rights and signed a standard consent form before the study. We issued a small token of appreciation as incentive for the study without monetary payments. For two of the first studies (P1 and P6), one of the authors of the paper was on the same call as an observer with the participant, and provided light clarification assistance when needed. We made minor changes to the instructions to ensure subsequent unmoderated studies can proceed smoothly. All studies were conducted during Nov 6 - 10, 2023.

The study was within-subject and counterbalanced. For each part of the study, participants were instructed to do a travel planning task to a destination of their choice, one using ChatGPT, the other using EXPLORELLM. We always instruct the participants to plan the trip to the same destination

in the second task as the previous one. We provided a login with ChatGPT plus subscription and participants were instructed to use ChatGPT-4². We randomized the order of the system the participants use. In the end, we had five participants using ChatGPT first (P1-5), and three participants using EXPLORELLM first (P6-8).

We collected age and gender as demographic information in the exit survey. We transcribed the user study videos and two authors of the paper conducted qualitative coding on the transcripts to identify common themes, first independently, and then discussed and agreed on the final findings.

3.3.1 Main Task. We provided the following instructions for the main task. It is important to note that our system design is not limited to travel planning. We chose to evaluate trip planning as an example task because it is general, typically complex, and allows for individual user constraints and preferences.

Key study instructions are as follows:

Imagine you are planning a trip to a destination of your choice using [system name].

- You have less than 5 min to complete this task.
- You can have as many interactions as you like using this system.
- Please make sure this trip is personalized to you (e.g., cost, time, location).
- Don't forget to think aloud as you try to complete the task!

Once you have completed the task, answer these following questions verbally:

- Tell us in detail, what do you find most helpful and unhelpful from this result?
- If at all, which part of the result do you find personalized to you?
- If at all, how much does this system make you feel more or less confident about planning a trip?
- Is there anything that you would like to comment about this task?

3.3.2 Participant Demographics. Eight participants took part in the study, out of which six identified as male, one female, and one preferred not to say. Seven out of eight participants are within the 25-34 years old, with one participant 35-44 years old. Seven participants are located in the U.S., with one in the U.K.. We did not collect other demographic information for privacy due to the limited sample size. We acknowledge that the lack of diversity of participants is a significant limitation, and discuss future work to address this problem in later sections.

4 RESULTS

Overall, participants confirmed our hypotheses that the current chatbot system provides generic and verbose responses, and that they liked EXPLORELLM's ability to provide structured task breakdown and personalization, despite some usability issues. Importantly, we also found that hallucination is a major hurdle in building trust in the system. Participants often pointed out where the information provided in the system is wrong, or that they don't trust the information and need to conduct their own research for additional verification. Finally, participants expressed wishes for more control of the system, richer content and tool use, which we discuss in future work.

Below we report in more detail each of the findings. We refer to a particular participant as P#.

4.1 Limitations of the Chat-Only UI

Participants pointed out two major limitations of the current chat-only UI for exploratory tasks. All but one participant mentioned that ChatGPT responses are verbose and generic. Multiple

²Note during the study period the ChatGPT system was changed from ChatGPT-4 to turbo. We do not know the details of the change but in ChatGPT-4 turbo, the system also has web browsing capability, which resulted in some of the participants commenting about transparency of the system.

participants used the “stop generating” button to interrupt ChatGPT mid-response (while the response is streaming), commenting *“this is really verbose”* (P5). P3 stated, *“I didn’t need this much text to start with.”* P7 said, *“too much, stop it”*. Often, people interrupt the generation when they notice the response being streamed back is off track. After the interruption, people often ask follow up questions to steer the conversation to another direction. For example, P5 asked *“Can you tell me when is the best time to go to [destination]?”* after noticing the ChatGPT was going on and on about the cost of the trip after the initial prompt.

The second major limitation of the chat-only interface is that responses are generic. P1, P2, P5 and P6 commented that ChatGPT gives generic answers. P1 stated, *“It was giving me kind of generic stuff. I had to remember to tell it stuff about me.”* P6 mentioned that *“It was not really personalized because this is just giving me information.”* Even when the user provides more personal preferences, sometimes ChatGPT fails to take it into consideration when generating responses. For example, P5 specifically mentioned something that they are not interested in but the system still generates responses containing those options, revealing a lack of steerability.

4.2 Structured UI and Guided Task Flow

Compared to the chat-only UI of ChatGPT, participants liked the structured UI of EXPLORELLM and the guided task flow.

In particular, participants found that the broken down list of sub-tasks was useful and helped them think, plan and navigate. P5 stated, *“Having these different aspects of the trip surface upfront is helpful [...] where was the best time, how do we get there, what are some things to do there? It’s great that the system surfaces these up front. So I don’t have to like consider all the different aspects because I feel like I will miss things.”* P2 stated, *“the logical flow was very clear.”* P3 said, *“the whole nature of guiding people through their problem is a big one.”*

People also liked the fact that the list of sub-tasks persists as a collection of artifacts to reference back on. Compared to the chat-only interface, P1 stated, *“It’s just nice not have a wall of text to have to look back through.”* In contrast to the wall of text of ChatGPT, people found EXPLORELLM’s UI glanceable and that it keeps a better record of the task state. P2 commented on the list of sub-tasks, saying that *“I really like the list [...] I can see those lists so I felt more confident.”* This is in contrast to the linear chat-only UI, where the users end up with a long chat history and have to *“dig back through to find the interesting parts”* (P1).

At the same time, users indicated their wish for more control. P1 indicated that *“I wish that I could [...] X out some of these things. [...] these seem pretty similar so I could like close one of those so I can remember [...] like keep the high priority ones open.”*

4.3 Personalization

Contrary to the generic responses in chat-only interactions, the personalization aspect of EXPLORELLM stood out to participants.

Participants found that it was easier to get personal preferences into the system because of the dedicated UI (P1, P2, P3, P6). P3 said, *“I do think this is more personalized because it was easier to get that out of me.”* P2 also noticed that the personalized context carries over globally throughout the entire interaction. *“In the middle of the tasks, I specified clean and budget and it carried over to other tasks, and it helped me to narrow down those options and prioritize.”*

Nonetheless, participants expect EXPLORELLM to be more proactive in eliciting their personal preferences, by doing *“more cognitive lift”*. When commenting on the options generated by the EXPLORELLM system for flights, P3 added, *“It should have probably asked me [...] where do I have airline miles [...] or am I a member of a thing? Do I have any preferences? I hate having to remember*

what to tell it. If I talk to a travel agent, I don't get that. [...] They know what to ask to start. So that's a pretty big difference."

4.4 Options as Responses

Five out of eight participants (P1, P2, P5, P6, P8) indicated liking having a list of options to choose from in EXPLORELLM. For some participants, the benefit mainly comes from reducing mental load. P8 said, *"I think the idea of specifying sort of vaguely what you're interested in, getting some options, picking one, and then having the other follow-up responses take those options into account make sense."* P2 added, *"It removed all the burdens and it'll is just basically gave me options that I just need to choose from so it was super helpful."* For others, such as P6, the benefit comes more from the ability to personalize. P6 saw different layover options for the flight in one of the sub-tasks, and pointed out that having different stopover options is important because sometimes there are visa limitations and they prefer to go through one of the options than other ones. Finally, people also liked the explainability of the options, indicating that the explanations associated with each option are helpful.

However, issues arise when there are similar or too many options. P5 observes that for the subtask "Check the best time to visit [destination]", *"the recommended and other options are pretty much the same"*. (The options were all in autumn, including late August, September, and early October.)

Notably, many participants expressed the desire for richer content (P5, P7) and tool use (P1, P2, P6). P5 suggested wanting to see the list of options on a map to visualize their relationships to each other and how close things are. *"Some richer content like images and Maps should help and if I could [...] click on this to [...] get more details about the champagne houses or click on the locations to understand more about [...] the scenery around it, that would be helpful."* P5 also switched tabs to search for some suggested destinations on Google maps mid-task. P7 also suggested that they want to see the full itinerary on a map.

P2 suggested connecting to external tools would be more helpful. When the system was loading, P6 speculated that the system was using some tools to search for flight options (the system was not using any tools yet and we discuss this in future work). Participants also expressed the desire for the system to take actions rather than providing information alone. P1 mentioned that even though the system provided information, *"I still have to do the annoying nitty-grittys like actually book the tickets and figure out calendar dates"*.

4.5 Hallucination

Across both ChatGPT and EXPLORELLM, participants noted hallucination as a major limitation and expressed reservations in trusting the results fully.

Participants noted multiple times that some information generated by ChatGPT and EXPLORELLM is wrong. P2 requested travel planning for the NeurIPS 2023 conference in EXPLORELLM. Although the system correctly breaks down the task into first looking into the location of NeurIPS conference, it tried to look up the location for 2022 rather than the latest. Further, the system hallucinates and thinks that the NeurIPS 2022 was in Vancouver, Canada. P2 was confused and looked up NeurIPS 2022's location, which is New Orleans, USA. Factual errors like this get propagated to downstream tasks such as hotel booking and is very disruptive to the entire experience. In another example, P6 noticed that one of the suggested airports for flight options did not exist. The EXPLORELLM system suggested one option to go from London to India as "Emirates EK6³ flight departs from London

³Fact checking shows that Emirates EK6 leaves from London Heathrow to Dubai.

Gatwick and arrives at Kolkata Bhawanipur with one stop at Dubai”. P6 commented that *“This is wrong, there is no airport at Bhawanipur.”*

Such hallucination has significant safety implications and also dramatically limits the system’s usefulness and the extent to which users trust the system to take actions autonomously. P1 commented on one of the ChatGPT responses, *“I don’t know if I would trust it even it’s telling me [...] be careful about the transit options being canceled because of snow, I think I need to do some research into exactly what that means”*. We discuss this further in the limitations and future work section, including opportunities for better grounding, tool use, and integration with other systems.

4.6 Usability Issues

As a research prototype, some usability issues have emerged in EXPLORELLM. Most notably, all participants noticed that the system has high latency. Almost all participants noted that the system is slow when generating different options. Because the response back from the prompts are structured json data, it is not very suitable for streaming, which could have reduced the perception of latency. As the underlying LLMs become faster, the latency issue can be mitigated to a large extent.

Providing more transparency about the process to the users can be another design solution for handling latency. When the system is loading, currently EXPLORELLM indicates “thinking”. P3 wondered, *“What is it thinking about? ChatGPT with browsing is a little bit clearer because it’s telling me what tools it’s using and how it’s thinking about stuff.”* As more tools and functionalities get integrated into the system, revealing the inner workings of the system is important for user transparency and trust.

5 DISCUSSION

In this work, we introduced the EXPLORELLM system designed to provide more structured task guidance and easier personalization. Echoing findings in recent work [14, 44], our user studies support the motivating hypothesis that current chatbots’ responses can be verbose and generic. Participants liked having the structure of sub-tasks and the ability to personalize, wishing for even more control.

One of the most important findings of our work is that much of the prompt engineering work and the “thoughts” of LLMs can have direct user benefits when appropriately exposed to the end users. This intuitively makes sense as many underlying structures of LLM reasoning methods are compatible with how humans think and solve problems. We show that prompt-based methods can effectively aid humans in creating schemata (or “mental models”) for problem solving. With the scaffold of a logical task structure, each sub-tasks can be explored separately to reduce cognitive load, while loosely coupled together for effective information coordination across the system.

In addition, our findings show that task decomposition is promising for better tool use. LLM assistants do not exist in isolation, and users wish for a tighter integration with existing data and tools. Tool use is especially important given that hallucination presented itself as a major hurdle in gaining user trust. Through task decomposition, we can break down a much more complex task into concrete sub-tasks that have readily made tools suitable for the sub-task (e.g., checking the weather of a particular location; searching for a particular type of place on maps).

We see that participants intuitively begin to expect or wish for better tool use once they start exploring sub-tasks. The tree-like nature of EXPLORELLM tasks system makes it highly compatible for deeper decomposition and integration with external tools. For example, one of the sub-tasks generated by the EXPLORELLM system for travel planning is flight booking. This sub-task can be further broken down into sub-sub-tasks, such as deciding on dates and specifying departure locations. Once the user interacts with the sub-sub-tasks, we can map user input to parameters when invoking specialized tools such as flight search engines to provide accurate and personalized

results. For complex open-ended tasks that require extensive planning, a related open problem is how to decide when to use tools autonomously to solve sub-problems, and when to elicit user feedback.

More generally, our work shows the promise of re-imagining the relationship between natural language user interfaces (NLUIs) and graphical user interfaces (GUIs) [13]. As much as people are excited about the giant leap in NLUIs, there lies an opportunity to re-imagine the design of GUIs under this new interaction paradigm. EXPLORELLM offers one exploration of the possibilities in creating new “hybrid” systems that integrate the best parts of both natural language and graphical user interfaces.

6 LIMITATIONS AND FUTURE WORK

One of the most significant limitations of our work is the lack of diversity in participants. In addition to convenience sampling, our participants have limited representation in terms of age and gender, and are biased towards experienced users of AI chatbots. Five of our participants reported interacting with AI chatbots a few times a week, and the rest three reported using it every day. We plan to expand participant diversity in a follow up work.

Secondly, we only explored one layer of task decomposition without tool use or data integration. Future work can extend to more layers of task decomposition and integrate existing tools to sub-tasks, or even explore leveraging the tool making abilities of LLMs itself [3, 42]. Our system only elicits user context through input text, and future work can explore pulling user contexts directly from user data and external apps, such as emails and calendar (e.g., what reservations users already made, what events are on the calendar, what are the dates of the travel). In addition, we only explored a checkbox UI and displayed options in sub-tasks as a list. Future work can explore richer UI as participants mentioned, such as maps and diagrams.

The prompt we used for task decomposition and options generation endpoints can be further tuned for quality and diversity. For example, we can tune both prompts to avoid repetition or options that are too similar. It is also important to consider fairness in options generation given prior work on algorithm fairness [6] and the impact on user behaviors by social media ranking algorithms [7] and to guard against overreliance [34]. While we included the prompt to generate “diverse and representative” options, we did not formally evaluate the diversity of options generated due to the limitation of sample size. Future work could formally evaluate the impact of prompts on the diversity and quality of options generated, and their downstream impacts on user behavior.

Finally, as a research prototype, our implementation still has usability issues. We plan to open-source our implementation following the publication of the manuscript to allow the community to build and iterate on EXPLORELLM.

7 ACKNOWLEDGEMENT

We thank Jim Maddock for designing the initial version of the user evaluation plan and literature review. We also thank Melvin Johnson and Varun Godbole for their insightful feedback during the early phase of this project.

8 AUTHOR CONTRIBUTION STATEMENTS

Xiao led the overall project, co-designed the personalization, options generation, and summarization interactions, implemented the system, and conducted user studies and analysis. Swaroop developed the original idea of tree-based interaction, came up with the initial design, demo video and write up. Ariel contributed to user studies and interaction design. Sophie iterated on the initial demo and designed the system UI and interaction. Jilin contributed to the paper framing and user studies design. Chinmay participated in early discussions about the LLM sub-task decomposition method,

and helped with theoretically grounding work. Heng-Tze and Quoc provided continuous advice and feedback on the research of prompt decomposition and tree-based user-LLM interactions. Ed provided feedback on paper framing and connection to prior literature. All authors reviewed the manuscript.

REFERENCES

- [1] Saul Albert and Jan P De Ruiter. 2018. Repair: the interface between interaction and cognition. *Topics in cognitive science* 10, 2 (2018), 279–313.
- [2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.
- [3] Tianle Cai, Xuezhong Wang, Tengyu Ma, Xinyun Chen, and Denny Zhou. 2023. Large language models as tool makers. *arXiv preprint arXiv:2305.17126* (2023).
- [4] Minsuk Chang, Stefania Druga, Alexander J Fiannaca, Pedro Vergani, Chinmay Kulkarni, Carrie J Cai, and Michael Terry. 2023. The Prompt Artists. In *Proceedings of the 15th Conference on Creativity and Cognition*. 75–87.
- [5] Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588* (2022).
- [6] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. 2017. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining*. 797–806.
- [7] Motahhare Eslami, Aimee Rickman, Kristen Vaccaro, Amirhossein Aleyasen, Andy Vuong, Karrie Karahalios, Kevin Hamilton, and Christian Sandvig. 2015. "I always assumed that I wasn't really that close to [her]" Reasoning about Invisible Algorithms in News Feeds. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. 153–162.
- [8] Kristie Fisher, Scott Counts, and Aniket Kittur. 2012. Distributed sensemaking: improving sensemaking by leveraging the efforts of previous users. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 247–256.
- [9] Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*. PMLR, 10764–10799.
- [10] Andreas Gegenfurtner, Erno Lehtinen, Laura Helle, Markus Nivala, Erkki Svedström, and Roger Säljö. 2019. Learning to see like an expert: On the practices of professional vision and visual expertise. *International Journal of Educational Research* 98 (2019), 280–291.
- [11] Prakhar Gupta, Cathy Jiao, Yi-Ting Yeh, Shikib Mehri, Maxine Eskenazi, and Jeffrey P Bigham. 2022. InstructDial: improving zero and few-shot generalization in dialogue through instruction tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 505–525.
- [12] James Hollan, Edwin Hutchins, and David Kirsh. 2000. Distributed cognition: toward a new foundation for human-computer interaction research. *ACM Transactions on Computer-Human Interaction (TOCHI)* 7, 2 (2000), 174–196.
- [13] Bernard J Jansen. 1998. The graphical user interface. *ACM SIGCHI Bulletin* 30, 2 (1998), 22–26.
- [14] Peiling Jiang, Jude Rayan, Steven P Dow, and Haijun Xia. 2023. Graphologue: Exploring Large Language Model Responses with Interactive Diagrams. *arXiv preprint arXiv:2305.11473* (2023).
- [15] Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406* (2022).
- [16] Yoonsu Kim, Jueon Lee, Seoyoung Kim, Jaehyuk Park, and Juho Kim. 2023. Understanding Users' Dissatisfaction with ChatGPT Responses: Types, Resolving Tactics, and the Effect of Knowledge Level. *arXiv preprint arXiv:2311.07434* (2023).
- [17] David Kirsh. 1995. Complementary strategies: Why we use our hands when we think. (1995).
- [18] Scott R Klemmer, Björn Hartmann, and Leila Takayama. 2006. How bodies matter: five themes for interaction design. In *Proceedings of the 6th conference on Designing Interactive systems*. 140–149.
- [19] Kirby Kuznia, Swaroop Mishra, Mihir Parmar, and Chitta Baral. 2022. Less is More: Summary of Long Instructions is Better for Program Synthesis. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 4532–4552.
- [20] Preeti Lahoti, Nicholas Blumm, Xiao Ma, Raghavendra Kotikalapudi, Sahitya Potluri, Qijun Tan, Hansa Srinivasan, Ben Packer, Ahmad Beirami, Alex Beutel, et al. 2023. Improving Diversity of Demographic Representation in Large Language Models via Collective-Critiques and Self-Voting. *arXiv preprint arXiv:2310.16523* (2023).
- [21] Michael Xieyang Liu, Jane Hsieh, Nathan Hahn, Angelina Zhou, Emily Deng, Shaun Burley, Cynthia Taylor, Aniket Kittur, and Brad A Myers. 2019. Unakite: Scaffolding developers' decision-making using the web. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 67–80.

- [22] Xiao Ma, Swaroop Mishra, Ahmad Beirami, Alex Beutel, and Jilin Chen. 2023. Let’s Do a Thought Experiment: Using Counterfactuals to Improve Moral Reasoning. *arXiv preprint arXiv:2306.14308* (2023).
- [23] Dor Ma’ayan, Wode Ni, Katherine Ye, Chinmay Kulkarni, and Joshua Sunshine. 2020. How domain experts create conceptual diagrams and implications for tool design. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–14.
- [24] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651* (2023).
- [25] Paul P Maglio, Teenie Matlock, Dorth Raphaely, Brian Chernicky, and David Kirsh. 2020. Interactive skill in Scrabble. In *Proceedings of the twenty-first annual conference of the cognitive science society*. Psychology Press, 326–330.
- [26] Sandra P Marshall. 1995. *Schemas in problem solving*. Cambridge University Press.
- [27] Marvin Minsky. 1974. A framework for representing knowledge.
- [28] Swaroop Mishra, Matthew Finlayson, Pan Lu, Leonard Tang, Sean Welleck, Chitta Baral, Tanmay Rajpurohit, Oyvind Tafjord, Ashish Sabharwal, Peter Clark, et al. 2022. LILA: A Unified Benchmark for Mathematical Reasoning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 5807–5832.
- [29] Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2022. Reframing Instructional Prompts to GPTk’s Language. In *Findings of the Association for Computational Linguistics: ACL 2022*. 589–612.
- [30] Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-Task Generalization via Natural Language Crowdsourcing Instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 3470–3487.
- [31] Julia Neidhardt, Rainer Schuster, Leonhard Seyfang, and Hannes Werthner. 2014. Eliciting the users’ unknown preferences. In *Proceedings of the 8th ACM Conference on Recommender systems*. 309–312.
- [32] Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. 2022. Show Your Work: Scratchpads for Intermediate Computation with Language Models. In *Deep Learning for Code Workshop*.
- [33] Douglas W Oard and Jinmook Kim. 1998. Implicit feedback for recommender systems. (1998).
- [34] OpenAI. 2023. GPT-4 Technical Report. *arXiv:2303.08774* [cs.CL]
- [35] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* 35 (2022), 27730–27744.
- [36] Mihir Parmar, Swaroop Mishra, Mirali Purohit, Man Luo, Murad Mohammad, and Chitta Baral. 2022. In-BoXBART: Get Instructions into Biomedical Multi-Task Learning. In *Findings of the Association for Computational Linguistics: NAACL 2022*. 112–128.
- [37] Pruthvi Patel, Swaroop Mishra, Mihir Parmar, and Chitta Baral. 2022. Is a Question Decomposition Unit All We Need?. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 4553–4569.
- [38] Sarah R Powell. 2011. Solving word problems using schemas: A review of the literature. *Learning Disabilities Research & Practice* 26, 2 (2011), 94–108.
- [39] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350* (2022).
- [40] Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2022. A Recipe for Arbitrary Text Style Transfer with Large Language Models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 837–848.
- [41] Swarnadeep Saha, Omer Levy, Asli Celikyilmaz, Mohit Bansal, Jason Weston, and Xian Li. 2023. Branch-Solve-Merge Improves Large Language Model Evaluation and Generation. *arXiv preprint arXiv:2310.15123* (2023).
- [42] Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761* (2023).
- [43] Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, et al. 2022. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations*.
- [44] Sangho Suh, Bryan Min, Srishti Palani, and Haijun Xia. 2023. Sensecape: Enabling Multilevel Exploration and Sensemaking with Large Language Models. *arXiv preprint arXiv:2305.11483* (2023).
- [45] Masaki Suwa, Barbara Tversky, John Gero, and Terry Purcell. 2001. Seeing into sketches: Regrouping parts encourages new interpretations. In *Visual and spatial reasoning in design*. 207–219.
- [46] Perry W Thorndyke and Barbara Hayes-Roth. 1979. The use of schemata in the acquisition and transfer of knowledge. *Cognitive psychology* 11, 1 (1979), 82–106.

- [47] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. *arXiv:2203.11171* [cs.CL]
- [48] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560* (2022).
- [49] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned Language Models are Zero-Shot Learners. In *International Conference on Learning Representations*.
- [50] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* 35 (2022), 24824–24837.
- [51] Terry Winograd. 1971. Procedures as a representation for data in a computer program for understanding natural language. (1971).
- [52] Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers. *arXiv preprint arXiv:2309.03409* (2023).
- [53] 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 preprint arXiv:2305.10601* (2023).
- [54] JD Zamfirescu-Pereira, Richmond Y Wong, Bjoern Hartmann, and Qian Yang. 2023. Why Johnny can't prompt: how non-AI experts try (and fail) to design LLM prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–21.
- [55] Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H Chi, Quoc V Le, and Denny Zhou. 2023. Take a Step Back: Evoking Reasoning via Abstraction in Large Language Models. *arXiv preprint arXiv:2310.06117* (2023).
- [56] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625* (2022).
- [57] Tamara Zubatiy, Niharika Mathur, Larry Heck, Kayci L Vickers, Agata Rozga, and Elizabeth D Mynatt. 2023. "I don't know how to help with that"-Learning from Limitations of Modern Conversational Agent Systems in Caregiving Networks. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW2 (2023), 1–28.

A APPENDIX

We provide some examples of interaction results with Google Bard (Figure 7), OpenAI ChatGPT (Figure 8) and EXPLORELLM (Figure 9, 10 and 11) for reference.

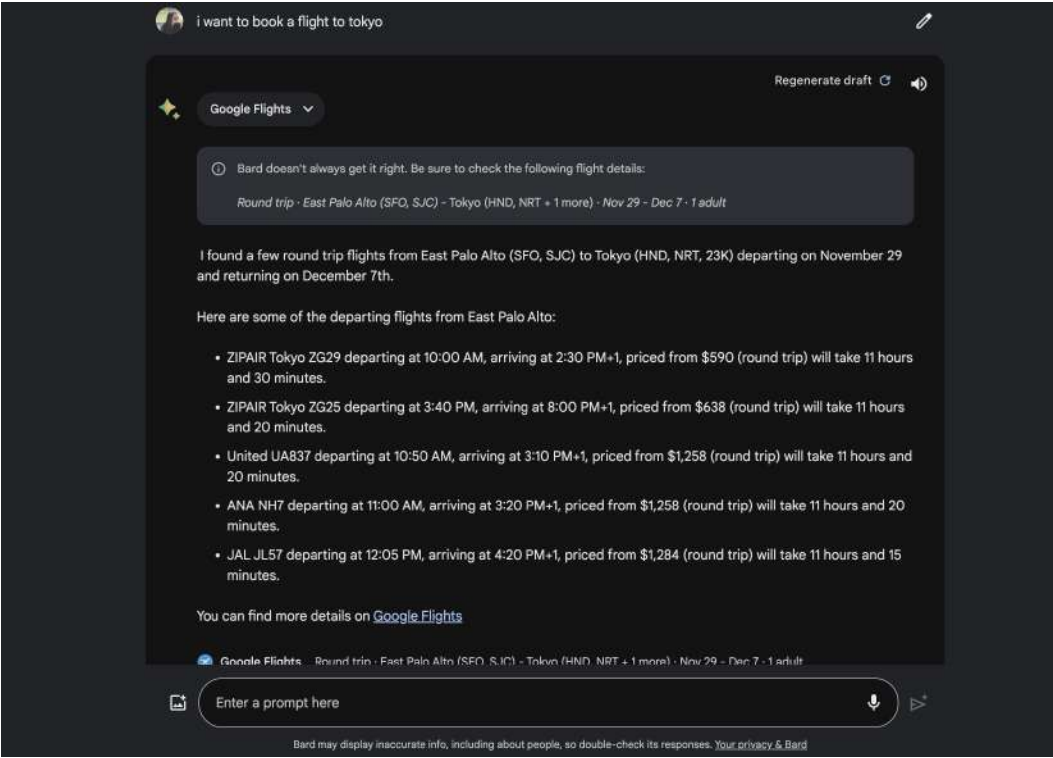


Fig. 7. Bard results for the query “I want to book a flight to Tokyo”

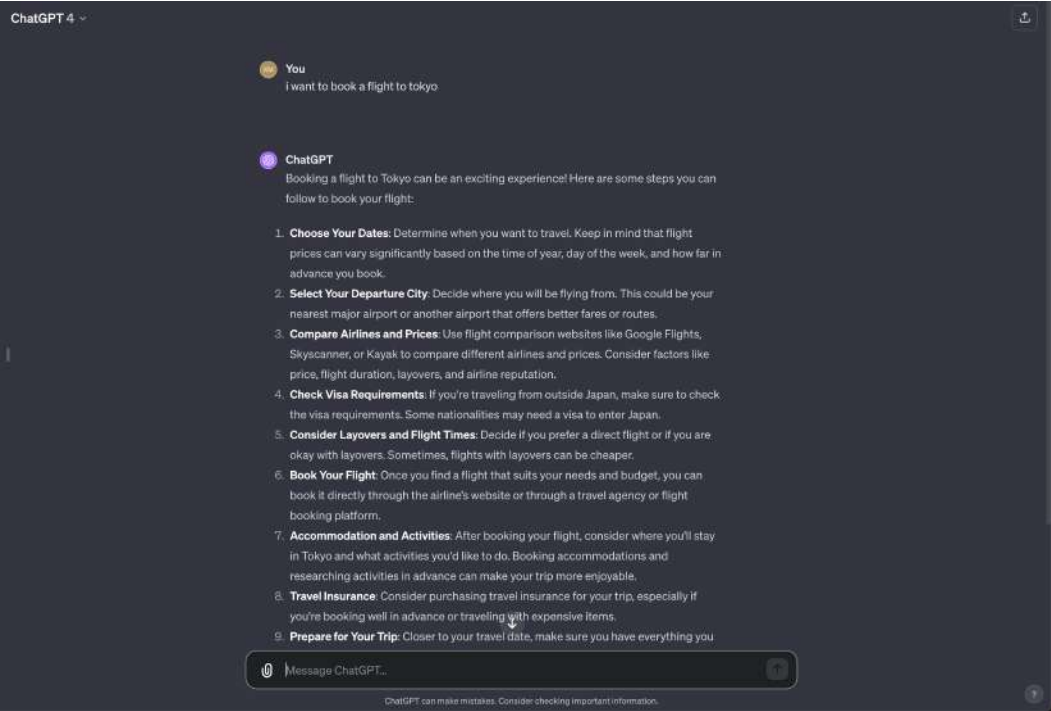


Fig. 8. GPT-4 results for the query “I want to book a flight to Tokyo”

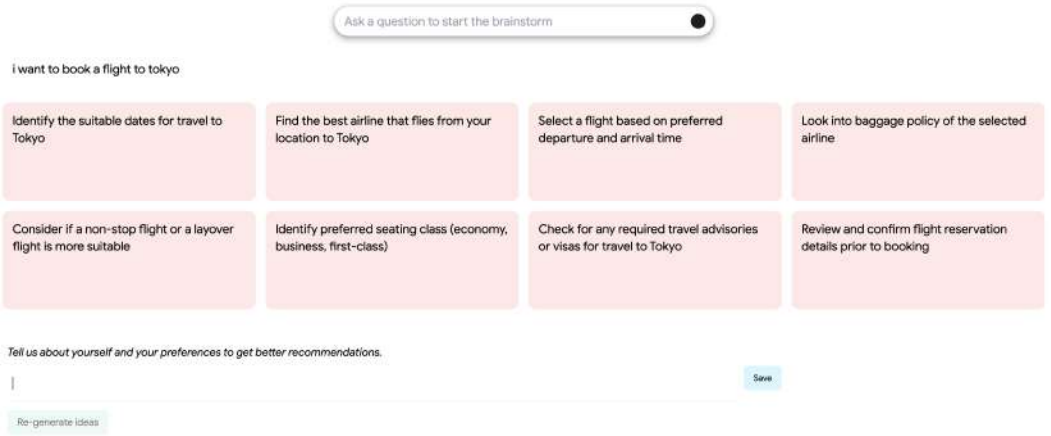


Fig. 9. EXPLORELLM results for the query “I want to book a flight to Tokyo”. Users can further interact with each of the sub-tasks.

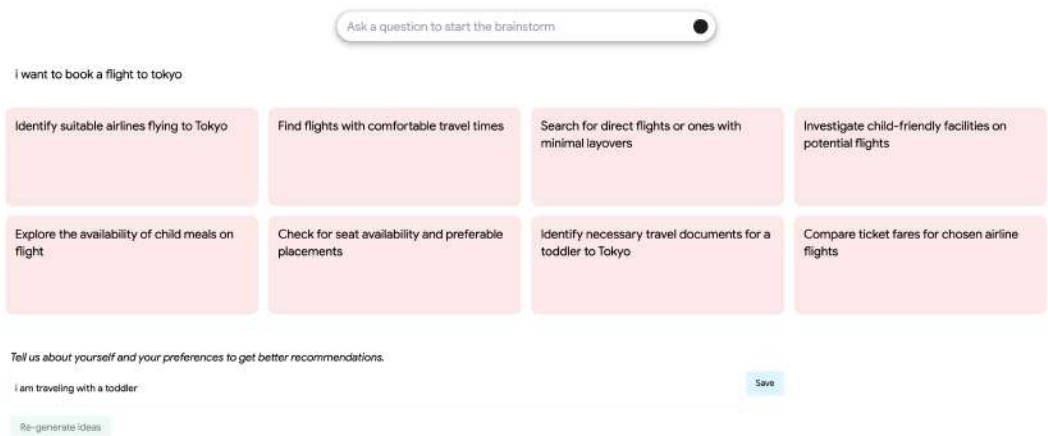


Fig. 10. EXPLORELLM regenerates and refreshes the sub-tasks for the query “I want to book a flight to Tokyo” when the user specifies “I am traveling with a toddler”.

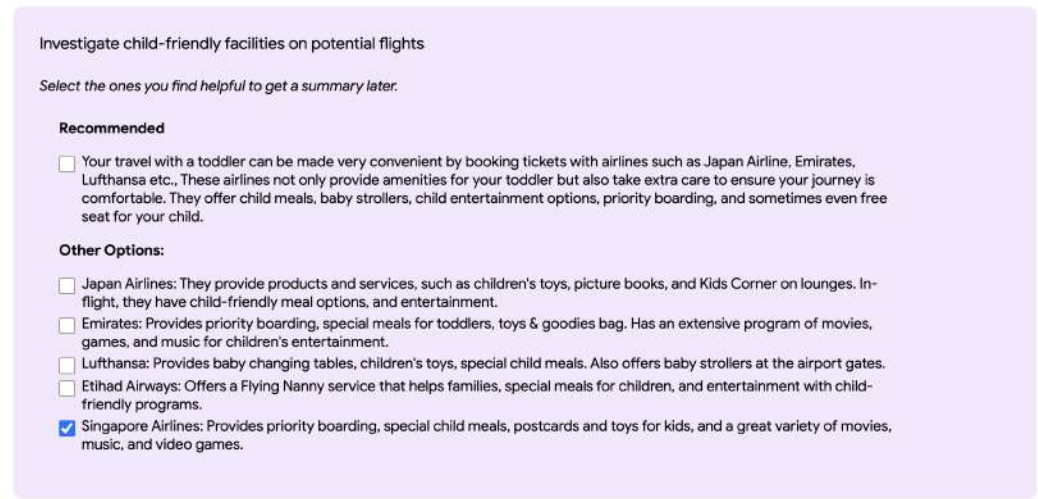


Fig. 11. When the user interacts with a subtask such as “investigate child-friendly facilities on potential flights”, EXPLORELLM generates a list of options (may contain hallucination).