ASME Journal of Computing and Information Science in Engineering
Online journal at:
https://asmedigitalcollection.asme.org/computingengineering

Elicitron: A Large Language Model Agent-Based Simulation Framework for Design Requirements Elicitation

Mohammadmehdi Ataei[1]
Autodesk Research,
661 University Avenue,
Toronto, ON M5G 1M1, Canada
e-mail: mehdi.ataei@autodesk.com

Hyunmin Cheong
Autodesk Research,
661 University Avenue,
Toronto, ON M5G 1M1, Canada
e-mail: hyunmin.cheong@autodesk.com

Daniele Grandi
Autodesk Research,
The Landmark @ One Market, Ste. 400,
San Francisco, CA 94105
e-mail: Daniele.Grandi@autodesk.com

Ye Wang
Autodesk Research,
The Landmark @ One Market, Ste. 400,
San Francisco, CA 94105
e-mail: ye.wang@autodesk.com

Nigel Morris
Autodesk Research,
661 University Avenue,
Toronto, ON M5G 1M1, Canada
e-mail: nigel.morris@autodesk.com

Alexander Tessier
Autodesk Research,
661 University Avenue,
Toronto, ON M5G 1M1, Canada
e-mail: alex.tessier@autodesk.com

Requirement elicitation, a critical, yet time-consuming and challenging step in product development, often fails to capture the full spectrum of user needs. This may lead to products that fall short of user expectations. This article introduces a novel framework that leverages large language models (LLMs) to automate and enhance the requirement elicitation process. LLMs are used to generate a vast array of simulated users (LLM agents), enabling the exploration of a much broader range of user needs and unforeseen use cases. These agents engage in product experience scenarios, explaining their actions, observations, and challenges. Subsequent agent interviews and analysis uncover valuable user needs, including latent ones. We validate our framework with three experiments. First, we explore different methodologies for the challenge of diverse agent generation, discussing their advantages and shortcomings. We measure the diversity of identified user needs and demonstrate that context-aware agent generation leads to greater diversity. Second, we show how our framework effectively mimics empathic lead user interviews, identifying a greater number of latent needs than conventional human interviews. Third, we show that LLMs can be used to analyze interviews, capture needs, and classify them as latent or not. Our work highlights the potential of using LLMs to accelerate early-stage product development with minimal costs and increase innovation.
[DOI: 10.1115/1.4067388]

Keywords: requirements elicitation, large Language models, artificial intelligence, LLM agents, computer-aided design

1 Introduction

Requirement elicitation (RE) sits at the core of successful product design, yet it remains a complex and resource-intensive endeavor. Traditional RE methods, like interviews, focus groups, and prototyping, are invaluable but have inherent limitations. These methods are often time consuming, may not fully capture the diversity of user perspectives, and may miss underlying needs that are difficult for users to articulate [1,2]. The consequences of inadequate requirements elicitation can be significant, ranging from design misalignment to compromised product adoption.

Recent advancements in large language models (LLMs) present new possibilities for automating and augmenting requirement elicitation. LLMs, having learned the patterns and complexities of human language from vast textual corpora, seemingly possess a remarkable capacity for natural language understanding [3]. This potential can be leveraged to construct a simulated environment where LLM agents *role-play* a variety of potential users. These agents can simulate distinct viewpoints, engage in product experience scenarios, and participate in user interviews aimed at identifying user needs. By doing so, these virtual users could potentially uncover a wide spectrum of user needs, preferences, and pain points that might otherwise remain hidden or unexplored.

There are reasons to believe LLMs could offer unique advantages for identifying diverse and latent needs. The core capabilities of LLMs lie in their flexibility and ability to perform different tasks related to natural language. By incorporating contextual elements, LLMs can grasp nuances and make inferences through in-context learning [3], leading to more accurate and insightful analyses of user needs. Because LLMs have been trained on a vast amount of data, they likely have been exposed to a diverse set of user needs for a particular product. During training, LLMs may have also picked up behavioral patterns and subtle language cues that hint at underlying—sometimes even subconscious—needs experienced by users. Moreover, they may be able to perform analogical reasoning to relate experiences and needs identified across different

[1]Corresponding author.
Manuscript received April 1, 2024; final manuscript received December 2, 2024; published online January 16, 2025. Assoc. Editor: Yan Wang.

products and uncover novel needs. However, a potential contradiction arises when considering the inherent nature of LLMs: They are trained to predict the most likely next word or sequence of words based on patterns in their training data [4]. On the surface, this focus on selecting the most likely outcome might seem at odds with the goal of uncovering diverse or latent needs. While LLMs are indeed trained to predict likely word sequences based on training data patterns [4], they generate responses by creatively combining and extrapolating from their training data, rather than simply retrieving and reproducing exact matches. This creative process, often labeled as "hallucination" when it produces factually incorrect information, is in fact the LLM's core strength for generating novel and diverse outputs. The key lies in how we direct and constrain this generative capacity for a particular purpose. Through careful prompt engineering, contextualization, and the use of techniques like adjusting sampling parameters (e.g., temperature or top-P), we can guide the LLM to explore beyond obvious associations and produce outputs that reveal diverse and latent needs.

This research presents a new LLM-based framework, called *Elicitron*, for automating and augmenting the RE process. In Elicitron, LLM agents are constrained to produce structured outputs that are relevant to and useful for the requirement elicitation process. Elicitron also employs techniques to create diverse user-representing agents and simulate product experiences through the *action, observation, challenge* steps, inspired by *chain-of-thought* reasoning [5]. Finally, by creating agents with specific roles, Elicitron can discover interesting user needs that may otherwise be difficult to obtain with human interviews.

Elicitron's ability to create diverse user agents can be leveraged to identify a diverse set of user needs. Because the process of creating and interviewing these agents is automated, Elicitron is highly scalable, unlike the traditional RE methods. We conducted an experiment to evaluate the capability of Elicitron to generate a diverse set of user needs and identified a context-aware generation method that maximizes need diversity.

In addition, we show how Elicitron can be applied to identify latent needs—unarticulated and unexpected user needs that strongly influence product desirability [6]—which are significantly difficult to obtain with the traditional RE methods. This can be achieved by either automatically or manually creating user agents with *empathic lead user* (ELU) roles [7]. We conducted a second experiment to demonstrate that Elicitron can generate a higher number of latent needs than human interviews. Finally, a third experiment was conducted to show that, given criteria and chain-of-thought reasoning, LLMs are capable of identifying and classifying latent needs from interview data.

2 Background

2.1 Large Language Models. LLMs are machine learning models that appear to exhibit the ability to understand, reason with, and generate natural language [3]. LLMs have been shown to engage in fluent conversations, translate languages, and write various types of creative content in different styles. The development of LLMs marks a significant step toward human-like agent capabilities within artificial intelligence. Beyond traditional applications, the use of LLMs is expanding into areas like software development, content creation, customer service, and mechanical design generation and understanding [8–13].

LLMs are trained using self-supervised learning. They are trained on vast amounts of text data to predict the next word or sequence of words in a sentence. This training process, along with architectural advancements like the transformer model [4], allows LLMs to grasp patterns, nuances, and context within human language, including learning implied meanings [3].

The knowledge acquired during training provides the foundation for LLMs to effectively engage in role-playing scenarios [14–16]. They can simulate various roles by using specific language patterns, word choices, and sentence structures learned from their training data. For example, if an LLM encounters a training dataset rich with technical communication, it can adapt its vocabulary and sentence structure to convincingly adopt the role of an engineer during role-playing sessions.

LLMs' capability to role-play makes them useful tools in requirements elicitation. They can simulate a diverse set of users, including empathic lead users, providing valuable insights into their product experience and needs.

2.2 Design Requirement Elicitation, Empathic Design, and Latent Needs. During the requirement elicitation phase of engineering design, empathy plays an important role in helping engineers better understand user needs [17,18] and consequently the design problem [19]. By interviewing, observing, and empathizing with users, designers can derive structured design requirements from unstructured feedback from product users, in a process called *empathic design* [20,21].

There are two types of design requirements or user needs: direct needs, which tend to be obvious to the customer and lead to incremental changes in a product, and latent needs, which may be nonobvious and difficult to uncover [22].

User interviews or observations may not reveal latent needs that consumers deem important in a final product [23]. Identifying latent needs early in the design process has been found to speed up the development process, and their discovery benefits the design engineer by providing insights into extreme use cases that might push the product to its limits [24–26].

Over the years, the design research community has experimented with various empathic design methods to improve latent need discovery. Hannukainen and Holtta-Otto used photo diary and contextual inquiry with disabled people to identify latent needs [20] for a mobile phone. Lin et al. elicited latent needs from ordinary users by simulating extraordinary situations, e.g. using a blindfold to simulate limited sight or oven mittens to simulate limited dexterity [7] during setting up a tent. Issa et al. prompted designers interpreting user interviews to "write a statement as someone with [specific experience]", in an attempt to bias the designer to be more empathetic with a lead user [27].

More recently, LLMs have been evaluated on design requirement understanding and knowledge extraction tasks, and used for design requirement analysis to guide novice designers towards a conceptual design solution [12,28,29]. Zhu et al. theorized how artificial intelligence and LLMs might be leveraged to support data-driven user studies for empathic design, and later expanded their work to investigate the use of LLMs for inferring product users" goals and "fundamental psychological needs'" [30,31]. Outside of the product design domain, Barandoni et al. evaluated the performance of LLMs in extracting travel customer needs from travelers' requests posted to a travel website [32].

While the design research community has identified empathy as an important component of design requirement elicitation [7,33–36], interviewing empathic lead users or observing them in user studies remains a time consuming and costly activity. The use of LLMs to simulate this process by creating and interviewing a diverse set of empathic lead user agents could address the gaps.

2.3 Metrics for Design Diversity. In the design methodology literature, design diversity is typically considered an extension of novelty from a set of designs. A novel design is often considered to be unique by the person who created it (psychological novelty) or more generally unique to the field (historical novelty) [37]. Evaluating the novelty of a design is a subjective task, and it is typical to leverage consensual assessment technique or similar methods that involve asking domain experts to rate designs on criteria such as novelty [38–41]. However, recent work in deep generative models, which leverages machine learning methods trained on large datasets of prior designs to create novel design solutions, has led to the development and adoption of computational novelty and diversity metrics. These include the convex hull volume and the mean distance to centroid metrics, which can be used to measure the average diversity of a whole set of designs [42–49]. In addition to those measures, we are also interested in measuring the diversity of possible clusters of design ideas, and not just the outliers; thus, the *silhouette score* typically used in cluster analysis [50] may also be applicable.

The *convex hull volume* is defined as the hypervolume of the smallest convex set that includes all of the samples. It has been used to measure the diversity in different disciplines, but is sensitive to outliers [42,43,51]. A larger convex hull volume indicates that, in the embedding space, the samples cover more space and are more diverse.

The *mean distance to centroid* of the embeddings is computed by averaging the distance of each sample to the centroid of the whole set [42,43,52,53]. A larger value indicates that the samples in the set are further from the centroid and thus are assumed to represent more diverse concepts. This metric is more suitable for more uniform distributions and is also sensitive to outliers.

In the context of clustering algorithms, the *silhouette score* is a metric used to calculate the performance of a clustering technique. Its value ranges from –1 to 1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. The silhouette score has been used to measure the diversity of recommendation systems, but has not been adopted in the field of design research [54,55].

3 Architecture of Elicitron

Elicitron is designed to closely simulate real-world requirement elicitation processes. The architecture comprises four distinct components mirroring the phases involved in gathering requirements (Fig. 1). Each component is discussed in detail below. To maintain structural consistency and prevent workflow errors, we employ a Pydantic model to constrain LLM outputs, followed by a schema validation step.

3.1 Agent Generation. A significant challenge in requirement elicitation (either with LLMs or traditional methods) is capturing a diverse range of user viewpoints. For this reason, our framework's initial step generates a diverse set of agents to simulate users within the elicitation process. This mirrors real-world practices where a wide variety of users are deliberately selected for RE studies.

The LLM is instructed to generate three elements for each user agent:

- *Name:* A label representing the user agent.
- *Description:* A description of the user characteristics.
- *Reasoning Chain:* A rationale for creating this agent.

The first two elements comprise the description of a user role. The reasoning chain aids in understanding the LLM's agent generation logic, a process similar to *chain of thought* [5].

We have employed the following agent generation methods:

3.1.1 Parallel Agent Generation. Here, the LLM receives N independent prompts to generate N user agents simultaneously. This method is advantageous for rapid creation of a large number of agents in parallel. However, due to the LLM's lack of awareness of other agents being generated, diversity may be limited as the model could produce similar agents.

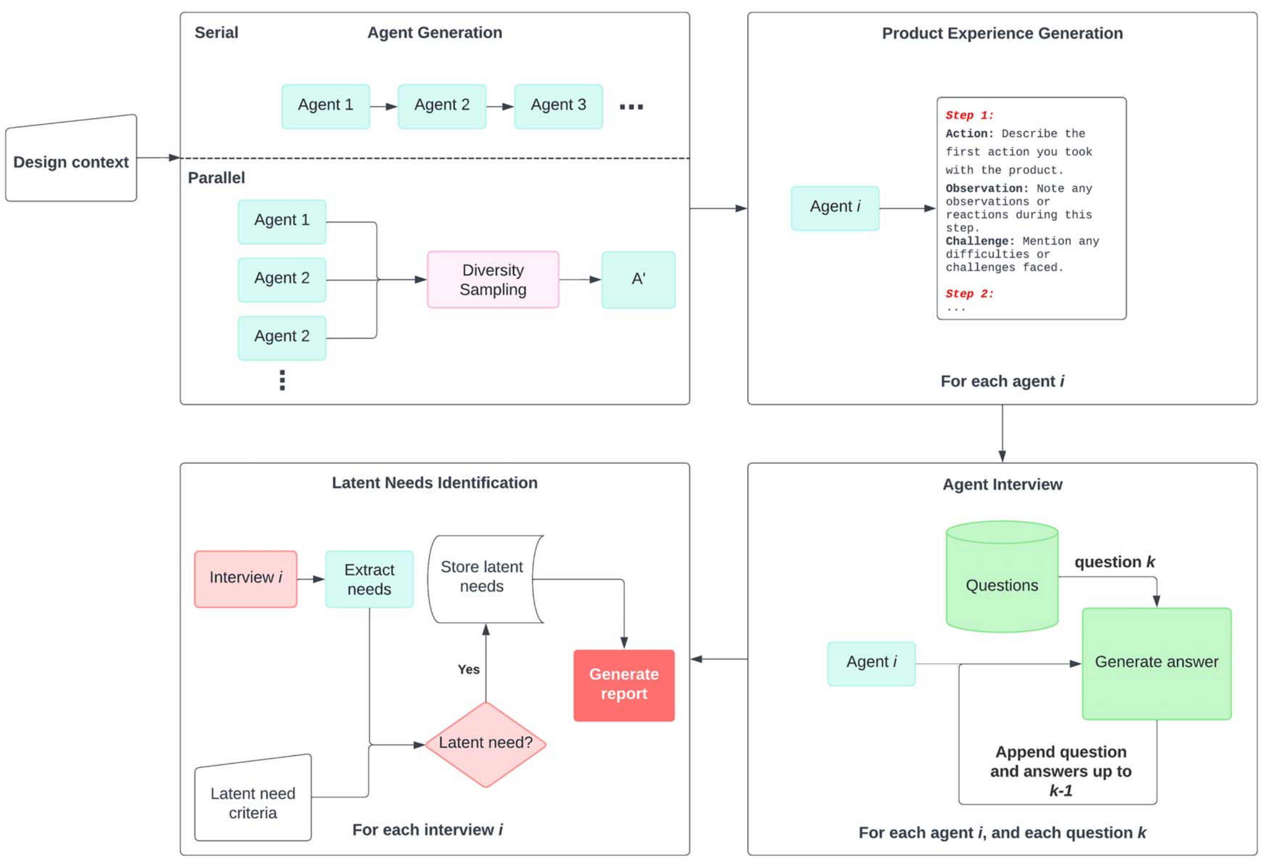

Fig. 1 Elicitron's architecture for requirements elicitation using LLMs: First, LLM agents are generated within a design context in either serial and parallel fashion (incorporating diversity sampling to represent varied user perspectives). These agents then engage in simulated product experience scenarios, documenting each step (Action, Observation, Challenge) in detail. Following this, they undergo an agent interview process, where questions are asked and answered to surface latent user needs. In the final stage, latent needs are identified using an LLM on a provided criteria, and finally a report is generated from the identified latent needs.

To mitigate this, we implement a *filtering* stage. We use a K-means clustering algorithm to group agents based on the similarity of their embeddings and then select only representative agents from each cluster to result in a diverse set of agents.

Given a set of generated agents $A = a_1, a_2, \ldots, a_N$, where each agent a_i is represented by an embedding vector v_i in a high-dimensional space, the goal is to select a diverse subset of agents.

(1) *Assign Embeddings:* First, assign an embedding vector v_i to each agent a_i description in the set A. We used `text-embedding-ada-002` by OpenAI for this purpose.
(2) *Perform K-Means Clustering:* Apply a clustering algorithm, K-Means(V, k). V is the matrix of all embedding vectors v_i and k is a chosen number of clusters ($k \leq N$). Given k, the K-means algorithm assigns each agent to the cluster with the nearest mean embedding.
(3) *Select Diverse Agents:* From each of the k clusters, select one representative agent. The selected agents are deemed diverse, as they come from different clusters in the embedding space. The resulting set of agents is denoted by A'.

This method may involve overgenerating agents, followed by filtering down to N agents. While we used K-means for filtering, other clustering techniques are also applicable.

3.1.2 Serial Agent Generation. In this technique, the LLM receives a single prompt to generate N agents. Here, the details of generated agents persist in the LLM's context to promote greater diversity compared to parallel generation. A drawback is that it runs more slowly than parallel generation. Moreover, there is a practical limit on the number of agents generated determined by the LLM's maximum token output length. At the time of this writing, most LLMs cap output at around 4096 tokens, which experimentally suggests a maximum of roughly 20 agents per call for each generation.

3.2 Product Experience Generation.

After generating a diverse agent pool, user agents are prompted to imagine their interaction with the potential product. This phase is essential for identifying specific usage scenarios that could lead to detailed and latent needs to be identified during the subsequent interview process.

(1) *Simulated Interaction:* Agents receive an open-ended prompt to describe steps they would take to interact with the product. This might involve setup, specific feature usage, or troubleshooting. Agents are allowed to explore freely, simulating the varied ways real users would interact with the product.
(2) *Structured Response Generation:* For each interaction step, agents provide responses organized into three elements:
   - *Action:* The description of the interaction step taken (e.g., setup, feature activation).
   - *Observation:* The agent's reactions and perceptions of the step. This includes both favorable impressions and points of friction.
   - *Challenge:* Explicit articulation of obstacles or difficulties encountered. This is done to uncover pain points in the user experience.

These structured responses serve as contextual foundations for the subsequent interview phase, effectively contextualizing the agent's experience. This approach mirrors the effectiveness of chain-of-thought prompting in generating more nuanced and in-depth responses.

An illustrative example of the product experience generation output can be found in Sec. 5.4. While it is not feasible to verify the factual accuracy of the output a priori, this methodology provides a structured framework that enhances the interpretability and reliability of the model's responses. Empirical studies, such as the work by Holbrook [56], have demonstrated the efficacy of scenario-based methodologies in improving the quality of elicited information.

3.3 Agent Interview.

The agent interview step mirrors real-world user interviews. It prompts each agent to reflect back on their product experience and asks follow-up questions aimed at uncovering user needs and nuanced insights identified from their product experience. The process works as follows:

(1) *Question Pool Creation:* A set of interview questions is prepared (human-developed or automatically generated by an LLM). These questions could be tailored to cover multiple aspects of the product, while also asking for innovative insights or improvement ideas.
(2) *Contextualized Questioning:* Questions are asked to each agent, integrating their prior Q&A responses and simulated product experiences into the LLM's context. This contextualization aims to alleviate the LLM's tendency to provide generic responses and facilitates the answers to be based on the individual agent's unique experience.

3.4 Latent Needs Identification.

In this workflow stage, we leverage previously collected interview responses to isolate needs automatically. An LLM processes agent interviews, extracting expressed needs. It then provides step-by-step reasoning for each identified need, drawing from established latent need criteria and examples, provided by the human experts. Finally, the LLM compiles all findings into a detailed report, offering insights on both expressed and latent needs uncovered during the analysis.

After the responses are collected from all user agents, the designer can review them to identify user needs that could be utilized for the subsequent design process.

In this work, we conducted three experiments to examine the value of Elicitron, all using *GPT-4-Turbo* from OpenAI as the LLM [57].

4 Experiment 1: Automatic Generation of Diverse Users and Their Needs

To examine the value of Elicitron in terms of identifying diverse user needs, we evaluated the agent generation methods proposed in Sec. 3. We generated 20 user agents each using three conditions, such as serial, parallel, and parallel with the K-means filtering, and compared the diversity of generated agents and their responses using computational metrics.

4.1 Evaluation of Diversity

4.1.1 Computational Evaluation. As discussed in Sec. 2.3, there is no de facto method to computationally evaluate the diversity of interview participants and their corresponding responses. Thus, we leveraged three different methods used in the prior work that rely on embedding the generated responses in a latent space to measure the diversity of design solutions: the convex hull volume [42,51], the mean distance to centroid [42,53], and the silhouette score [50].

For all three metrics, we first generated embeddings for each of the role descriptions and responses to 12 interview questions using *text-embedding-ada-002*. Note the interview questions asked are presented in Section 5.1. This resulted in 13 sets of 20 embeddings (for the user role and for each of the 12 questions). After using each of our three methods to generate the data ("serial," "parallel," and "parallel with filtering"), we computed the *convex hull volume* and the *mean distance to centroid* from the individual sets of 20 embeddings and normalized the results from 0 to 1.

The *silhouette score* is defined as follows. If a is the mean intracluster distance (the average distance between each point within a cluster) and b is the mean nearest-cluster distance (the average distance from a point to the nearest cluster of which it is not a part), then the silhouette score s for a single sample is given by the formula $s = \dfrac{b-a}{\max(a, b)}$. The mean silhouette score for a set of samples is the mean of the individual silhouette scores for each sample. An appropriate number of k clusters can then be selected by choosing the k cluster with the highest mean silhouette score, indicating that each cluster is very compact and distinct from other clusters. We used this metric both to choose a number of clusters for K-means, as well as a standalone metric to quantify the diversity of the samples for each of our methods.

4.1.2 Qualitative Evaluation. To further understand the differences along the diversity, we evaluated the content of the agent role descriptions and interview responses as follows:

(1) Cluster the embeddings of all 60 user agents generated from the three conditions using K-means. k is chosen based on the silhouette score to maximize distinct clusters.
(2) Summarize the k clusters of agents with the LLM with the following prompt, "Here are k groups of users, give a theme for each group. Group 1: $\cdots$" For the content of each group, the role descriptions of the agents in each cluster are used.
(3) Examine the coverage of clusters for each condition using scatter plots after reducing the dimensions with t-distributed stochastic neighbor embedding (t-SNE) [58].

4.2 Results

4.2.1 Convex Hull. Table 1 shows that the convex hull volumes were higher for the serial method than the parallel and parallel with filter methods, indicating a significant increase in diversity. Also, on average, the filtering method improved the diversity of the parallel generation method.

4.2.2 Mean Distance to Centroid. Table 2 shows the mean distance to centroid values. The results show that the serial method led to more diversity than the parallel and parallel with filtering methods, but by a smaller margin than the convex hull metric. Again, the "filtering" method improved the diversity of the "parallel" generation by a small amount.

4.2.3 Silhouette Score. We expect that a high silhouette score across various k's would indicate that the points are easier to cluster, or closer to each other, while a low silhouette score would indicate that the points are further apart, less clustered, and thus would represent a more diverse set. Figure 2 shows the computed silhouette scores for the three-generation methods. We can infer once again that the serial method produced the most diverse agents, compared to the parallel and parallel with filtering methods. There was no clear distinction between the "parallel" methods.

Table 1 Convex hull volumes of the embeddings of user role/descriptions and responses to interview questions for each generation method, normalized from 0 to 1

	Serial	Parallel	Parallel + filtering
User	0.991878	0.097886	0.081218
Characteristics	0.928448	0.191309	0.318411
Size	0.717999	0.382065	0.581811
Shape	0.929433	0.247194	0.273950
Weight	0.789456	0.314839	0.526912
Material	0.769544	0.310822	0.557846
Safety	0.659648	0.405045	0.633090
Durability	0.910748	0.347797	0.222656
Aesthetics	0.944532	0.152274	0.290983
Ergonomics	0.861179	0.338084	0.379566
Cost	0.910723	0.172481	0.375278
Setup	0.925232	0.292025	0.242214
Transport	0.944723	0.222418	0.240892
Mean	0.867965	0.267249	0.363448

Note: A higher convex hull volume indicates a relatively more diverse set.

Table 2 Mean distances to the centroid of the embeddings of user role/descriptions and responses to interview questions for each generation method, normalized from 0 to 1

	Serial	Parallel	Parallel + filtering
User	0.660156	0.527555	0.534677
Characteristics	0.618368	0.542512	0.568596
Size	0.590934	0.552930	0.587423
Shape	0.618861	0.551452	0.559385
Weight	0.610335	0.543569	0.576215
Material	0.601980	0.546090	0.582585
Safety	0.584500	0.562831	0.584449
Durability	0.623426	0.569565	0.535664
Aesthetics	0.649814	0.521597	0.552883
Ergonomics	0.614401	0.561247	0.554539
Cost	0.622532	0.536028	0.570199
Setup	0.632032	0.552557	0.543338
Transport	0.633343	0.543400	0.550992
Mean	0.620052	0.547026	0.561611

Note: A higher mean distance indicates a relatively more diverse set.

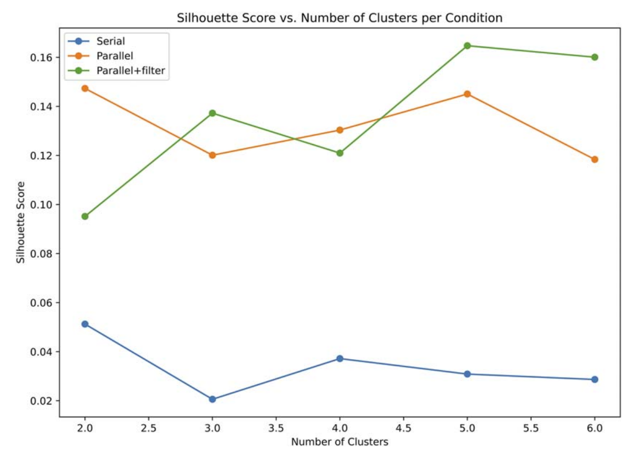

Fig. 2 The silhouette score measures the intracluster and intercluster distance. The serial method results in stakeholder embeddings that are more difficult to cluster compared to the parallel and parallel with filtering methods, which indicates that the serial embeddings are more diverse.

4.2.4 Qualitative Evaluation. We qualitatively examined the categories of users created for each condition. We aggregated all 60 users' embeddings created from the three-generation methods. We then chose $k = 4$ based on the silhouette score, i.e., when $k = 4$, it gives the most distinct clusters. By using K-means with four clusters, we found the following groups of users, as shown in Fig. 3:

(1) *Service and Conservation:* Roles including military, humanitarian work, and field research.
(2) *Outdoor Recreation and Camping:* Roles with activities such as minimalist camping and stargazing.
(3) *Adventure and Exploration:* Roles with activities such as hiking, backpacking, and mountain climbing.
(4) *Family Camping and Outdoor Activities:* Roles emphasizing bonding through camping and nature activities.

It should be noted that neither the parallel nor the parallel with filtering method created any user roles belonging to the service and conservation groups.

4.3 Discussion. From both computational and qualitative evaluation, the serial generation method led to the most diversity in the

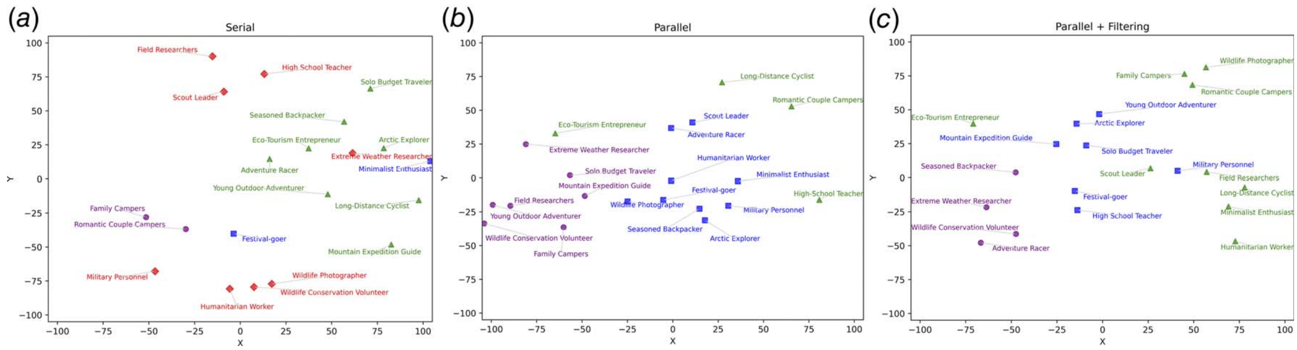

Fig. 3 Comparison of four groups of users' embeddings after reducing dimensions to 2 using t-SNE. Group 1: Service and conservation (diamond). Group 2: Outdoor recreation and camping (square). Group 3: Adventure and exploration (triangle). Group 4: Family camping and outdoor activities (circle). The serial generation gives the best coverage of all four groups. Parallel generation with and without filtering both missed service and conservation-related users.

user roles and the responses to interview questions. The parallel generation method with K-means filtering helps improve the diversity than using parallel generation only, albeit not significantly. The serial generation method benefits from maintaining previous agent generations as additional context to the LLM, allowing the LLM to avoid generating repetitions.

5 Experiment 2: Automatic Generation of Latent User Needs

In the second experiment, Elicitron was employed to automatically generate latent needs, which are difficult to identify using traditional interviews with human users.

To evaluate our method, the tent design example from Ref. [7] along with their reported results was used. In particular, we set our base condition using the ELU interview technique. This involves simulating extraordinary lead user conditions with regular users and interviewing them as the baseline condition. We then compare the number of latent needs identified with the ELU interview technique versus using three different conditions of our requirements elicitation method.

5.1 Experiment Setup. We set up three conditions to test the effectiveness of Elicitron. Each condition generated 20 user agents, the same number of people interviewed in Ref. [7].

- *Condition 1:* Automatic creation of user agents with the serial method
- *Condition 2:* Automatic creation of user agents with the serial method and addition of a steering prompt
- *Condition 3:* Manual creation of ELU agents

The serial method is used because it has been shown in Experiment 1 to create more diverse user agents. For condition 2, we provided the following additional prompt to encourage the creator agent to generate ELU agents. The text inside the double quotes is verbatim from Ref. [7].

> You must create nontypical users based on the following description of a typical user: "The typical user would be a weekend camper, 15-30 years old, with very good health and physical fitness, who camps a few times a year. The typical usage environment would be a public park or wilderness area, in a generally wooded or grassy environment with warm, sunny weather."

For condition 3, we manually created ELU agents based on the deviations from the experiences of a typical customer in a typical application and usage environment listed in Ref. [7]. The list of all user agents generated for each condition is shown in Table 3.

For all three conditions, we prompted the agents to engage in simulated product experience scenarios. We then asked LLM agents the following interview questions in sequence. These are the same interview questions used in Ref. [7] with human subjects,

Table 3 List of user agents generated for each condition

Condition 1 (automatic)	Condition 2 (automatic with steering)	Condition 3 (manual ELUs)
Young outdoor adventurer	Adventure-seeking teen	Outdoor enthusiast in the mountains
Family campers	Retired nature enthusiast	Hunter
Seasoned backpacker	Person with physical disability	Camper at desert canyons
Festival-goer	Winter camper	Professional mountaineer
Military personnel	Expedition leader	Professional rock climber
Romantic couple campers	Urban digital nomad	Pre-teen camper
Wildlife photographer	Rainforest explorer	Elderly with arthritis
Field researchers	High-altitude climber	Motion challenged teenager
Solo budget traveler	Family camping enthusiast	Visually impaired
Mountain expedition guide	Emergency preparedness advocate	Hearing impaired
Adventure racer	Festival-goer	Biologist
Scout leader	Field researcher	Financially challenged
Eco-tourism entrepreneur	Pet-loving camper	Parent with young children
Extreme weather researcher	Urban activist	Jungle trekker
Long-distance cyclist	Van life enthusiast	Summer arctic explorer
Humanitarian worker	Humanitarian worker	Amputee camper
High school teacher	Outdoor educator	Wheelchair accessible camper
Arctic explorer	Solo backpacker	Beach camper
Minimalist enthusiast	Eco-conscious camper	Back-country portage camper
Wildlife conservation volunteer	Outdoor sports organizer	Ultramarathon runner

but with some modification in wording to encourage the agents to provide specific needs and insights related to the question.

- *Free style*: "If you were to purchase an ideal tent, what main characteristics would you look for?"
- *Categorical*: "Focusing specifically on the [category], aspect of tent, can you tell me your needs and any innovative insights to address those needs?"
  Categories: size, shape, weight, material, safety, durability, aesthetics, ergonomics, cost, setup, transport.

5.2 Latent Needs Labeling. The responses given by the user agents were analyzed to identify the number of latent needs suggested by each agent. Again, we followed the criteria used by Ref. [7] to label whether a particular phrase was a latent need:

> If a reported customer need represented a significant change to the product design and did not match the categories [used in interview questions], then it was labeled as a latent need. Latent needs were also identified when a reported customer need represented an innovative insight into the product and/or product usage conditions.

Two raters performed the labeling task. Because determining a latent need is highly subjective and context dependent, the labeling was performed as follows. We began by randomly selecting 10% of the dataset for independent labeling by both raters. Following a calibration discussion to resolve discrepancies, we randomly selected another 20% of the data to calculate the interrater agreement score. Finally, the remaining 70% of the dataset was divided equally between the raters for independent labeling.

Because identifying latent needs from a free text is equivalent to an information retrieval task, an F-score is used to measure the interrate agreement as suggested by Ref. [59]. The F-score is computed as follows:

$$F_1 = 2\text{tp}/(2\text{tp} + \text{fp} + \text{fn}) \tag{1}$$

If both raters agreed on a particular phrase in the text as a latent need, it was counted as a true positive, tp. If the first rater identified it as a latent need but not the second rater, it was counted as a false positive, fp. If the second rater identified it as a latent need but not the first rater, it was counted as a false negative, fn. This choice of which rater's classification to consider as a positive is arbitrary because only the sum of fp and fn contribute to the denominator term in (1). We found the F-score of 0.83 (tp = 109, fp = 21, fn = 24), which indicated reliable agreement between the raters.

5.3 Experiment Results. The results (Fig. 4) show that the number of latent needs identified was higher for all three Elicitron conditions compared to the baseline, demonstrating the potential of our LLM-based requirements elicitation framework in identifying latent needs. Because the prior work [7] reported the mean number of latent needs but not any measure of variance, we could not conduct any statistical test to show statistical significance.

One-way analysis of variance was conducted to examine the effect of the agent creation conditions on the number of latent needs identified, but no significant difference was found, $F(2, 57)$, $= 2.3744$ $p = 0.1022$. The fact that the difference in the average number of latent needs between conditions 1 or 2 (automatic creation) versus condition 3 (manual creation) was not statistically significant indicates that the automatic creation methods can be just as effective as the manual creation method that requires domain knowledge. In addition, the lack of significant difference between conditions 1 and 2 indicates that a steering prompt may not be necessary. In fact, many of the user agents created in conditions 1 and 2 are similar to the ELU agents manually created in condition 3, as shown in Table 3. Examples include those who are outside of the typical ages ("adventure-seeking teen," "retired nature enthusiast"), those who are physically challenged ("person with physical disability"), and those facing extraordinary conditions ("expedition leader," "high-altitude climber"), etc.

While the latent needs identified with condition 1 did not significantly vary against the two other conditions, an interesting observation can be made regarding the individual user agents. Table 3 shows that many of the agents created do not necessarily represent ELU agents. For example, while user agents such as "adventure racer," "eco-tourism entrepreneur," or "high school teacher" could be considered unique user types, they did not entail extraordinary usage conditions and therefore led to the identification of relatively few latent needs (6, 3, and 4, respectively).

5.4 Example Outputs From Large Language Model Agents. Example outputs produced by LLM agents in the experiment are shown here. First, the product experience generated for a user agent titled "Elderly With arthritis":

Step 1:

- Action: Attempted to grasp the zipper of the tent's entrance with limited finger dexterity.
- Observation: The zipper was too small and lacked grip, making it difficult to manipulate.

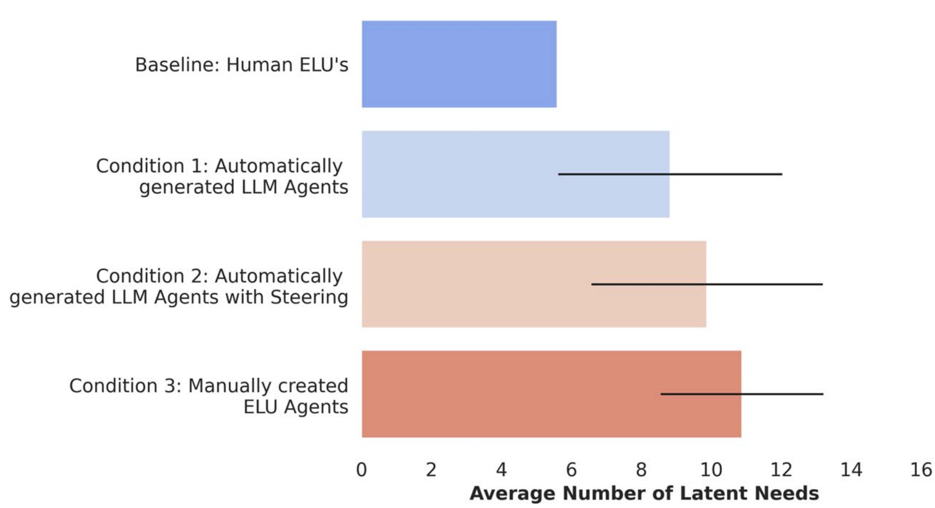

Fig. 4 Comparison of the average number of latent needs identified by each user agent across the experimental conditions. The error bars indicate standard deviation with $n = 20$ for each condition.

- Challenge: Faced significant struggle to open and close the tent, leading to frustration.

Step 2:

- Action: Tried to assemble the tent poles and stretch the fabric over the structure.
- Observation: The effort required to connect the tent poles and stretch the fabric exacerbated my arthritis pains.
- Challenge: Experienced severe difficulty due to lack of strength and pain in fingers, making assembly a painful and time-consuming process.

Step 3:

- Action: Attempted to stake the tent to the ground.
- Observation: Found the standard stakes and hammering method cumbersome and impossible to manage with my condition.
- Challenge: Couldn't effectively secure the tent to the ground, raising safety concerns in windy conditions.

Step 4:

- Action: Tried to pack away the tent after use.
- Observation: Struggled to fold the tent and roll it tightly enough to fit back into its bag.
- Challenge: The process was physically demanding and required more dexterity and strength than I possess, leading to reliance on others for assistance.

Here are example interview responses given by the same agent:

Q: If you were to purchase an ideal tent, what main characteristics would you look for?

- "The ideal tent would need to have large, easy-grip zippers designed for individuals with limited finger dexterity. It would include pre-assembled or quick-assembly poles that don't require significant strength or intricate manipulation, possibly with a simple click-together mechanism. Additionally, the tent should feature a staking system that doesn't rely on traditional hammering, perhaps using twist-in stakes that are more arthritis-friendly."

Q: Focusing specifically on the setup aspect of tent, can you tell me your needs and any innovative insights to address those needs?

- "…a self-erecting tent structure where the tent automatically expands and sets itself up upon being removed from its bag, eliminating the need for manually connecting tent poles or stretching fabric. This could leverage spring-loaded or memory material technology, where the structural elements are designed to automatically assume the correct form and tension when unleashed."

Finally, here are some examples of interesting latent needs identified from the experiment:

User: Visually impaired

- "This could mean a tent with a base that subtly slopes down towards the door, paired with a distinctive tactile path on the floor that leads directly to the entrance/exit."

User: Wheelchair accessible camper

- "All tent controls, such as zippers, vents, and lighting, should be within easy reach from a seated position."

User: Hunter (needs to set up a tent in dark)

- "…integrates a temporary, battery-powered LED guidance system. This system would activate upon initiating the setup process, illuminating each component in sequence (e.g., poles, connectors, and fabric) and guiding the user through the steps for assembly."

User: High-altitude climber

- "For enhanced stability in diverse conditions, the development of an adaptive anchoring system that automatically adjusts tension in response to wind and snow conditions could be revolutionary."

User: Outdoor sports organizer

- "A modular design ... would allow for connecting multiple tent units easily to expand the covered area ...the incorporation of a seamless interlocking system that enables tents to be connected without gaps or weak points."

6 Experiment 3: Automatic Detection of Latent User Needs

Analyzing interviews to detect latent needs is a challenging and time-consuming task that requires a deep understanding of the product and the customer's requirements. It can consume valuable resources and may not always yield consistent results across different analysts.

6.1 Experiment Setup. In this experiment, we evaluate the performance of LLMs in automating the analysis of interview texts and detecting latent needs. To facilitate this, we create a dataset consisting of 20 latent needs and 20 nonlatent needs, based on the evaluations of human experts in Experiment 2. This dataset will be used to assess the LLM's ability to accurately identify latent needs.

We conduct the evaluation using three different approaches:

(1) *Zero-Shot detection:* The LLM is tasked with labeling latent needs without any additional context, simply by answering the question "Is this a latent need?" with a binary response (True or False). There is no extra information or criteria given. The LLM relies solely on its existing knowledge to answer "true" or "false."

(2) *Detection With Latent Needs Criteria:* In this approach, the LLM is provided with the criteria to evaluate latent needs. Given these criteria, the LLM is asked to answer the question "Is this a latent need?" with a binary response (True or False). The criteria are as follows (adopted from Ref. [7]), which are the same criteria used by the human evaluators in experiment 2:

"Label the reported customer need as a latent need (latent = True) if either of the following conditions is met:
(a) The need represents a significant change to the product design and does not fall into any of the following categories: size, shape, weight, material, safety, durability, aesthetics, ergonomics, cost, setup, or transport.
(b) The need reflects an exceptionally innovative and clearly expressed insight regarding the product and/or how it is used."

(3) *Detection with Latent Needs Criteria and Chain-of-Thought:* In this final approach, the LLM is provided with the same criteria for latent needs as in the previous evaluation. However, the LLM is also instructed to use chain-of-thought analysis and think step-by-step to detect the latent needs. Given the criteria and the output of the chain-of-thought, the LLM answers the question "Is this a latent need?" with a binary response (True or False).

6.2 Experiment Results. Figure 5 presents the confusion matrices for the three evaluations, while Table 4 shows the corresponding performance metrics. In the zero-shot detection scenario, the LLM achieves a precision of 0.7273, recall of 0.8000, and an F1-score of 0.7619. These metrics indicate that the LLM can identify latent needs with reasonable accuracy

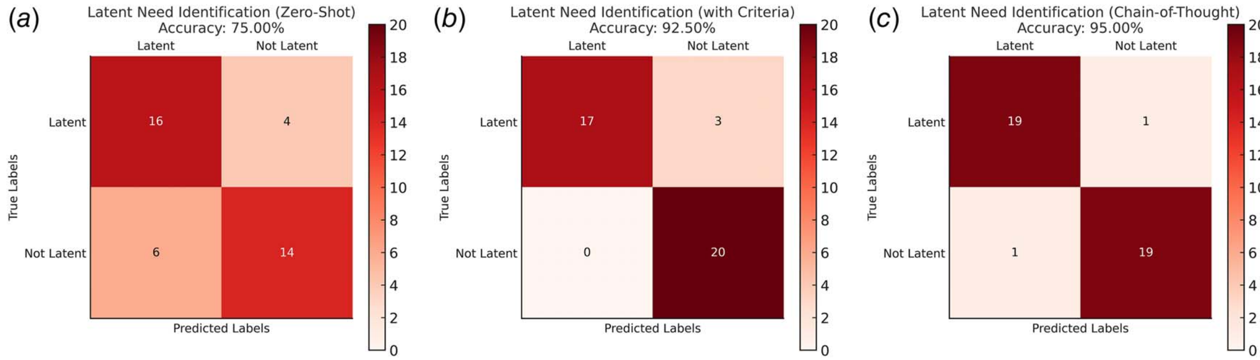

Fig. 5 Comparative confusion matrices for latent need identification: (*a*) zero-shot classification, (*b*) classification with latent need criteria, and (*c*) classification employing a chain-of-thought approach and latent need criteria

Table 4 Precision, recall, and F1-score for confusion matrices

Matrix	Precision	Recall	F1-score
Zero-shot	0.7273	0.8000	0.7619
With criteria	1.0000	0.8500	0.9189
Chain-of-thought	0.9500	0.9500	0.9500

purely based on their internal representations, even without any additional context. However, there is room for improvement, as the model struggles to distinguish between latent and nonlatent needs in several cases.

When provided with the latent needs criteria, the LLM's performance improves significantly. The precision reaches 10,000, indicating that all the needs identified as latent by the LLM are indeed latent needs. The recall increases to 0.8500, suggesting that the LLM can identify a higher proportion of the actual latent needs in the dataset.

The most impressive results are observed when the LLM is instructed to use chain-of-thought analysis in combination with the latent needs criteria. In this case, the precision, recall, and F1-score all reach 0.9500. This indicates that the LLM can accurately identify latent needs while minimizing both false positives and false negatives.

The observed improvements in the LLM's performance across the three scenarios are logical and expected. Each successive approach provides the model with increasingly relevant context and guidance, enabling it to better address the task at hand.

First, by providing the latent needs criteria, the LLM is able to focus its attention on the specific aspects that determine whether a need is latent. This targeted context helps the model zero in on the most pertinent information for making its determination.

Building upon that, the chain-of-thought analysis takes things a step further by guiding the LLM through a structured reasoning process. By breaking down the analysis into a series of smaller, interconnected steps, the model is able to systematically consider each criterion and build a logical argument for its ultimate conclusion. This approach helps ensure the LLM's output is well reasoned and grounded in the provided criteria.

The effectiveness of this approach is illustrated in Table 5, which presents two examples of the LLM's chain-of-thought reasoning for a latent and a nonlatent need. In each case, the model's step-by-step analysis clearly demonstrates how it arrived at its determination by carefully considering and applying each of the latent needs criteria in turn.

Beyond just improving performance, the use of chain-of-thought reasoning also enhances the interpretability of the model. Chain-of-thought opens a window into the LLM's decision-making process and makes it possible to understand the apparent logic behind each of the model's conclusions. This transparency is valuable in building trust and confidence in the model's outputs, as it allows users to verify that the LLM's decisions are based on sound logic and adhere to the specified criteria.

Table 5 Examples chain-of-thought reasoning by the LLM for latent need identification

Latent need	Nonlatent need
Interview: For the interior, a design focused on wide angles would maximize the field of view for photography through designated openings, enhancing the potential for wildlife observation without the need to leave the tent.	*Interview:* Regarding the durability aspect of tents, my primary need is for a tent floor that is resilient against tears from sharp objects and general wear.
LLM's chain of thought: The customer need specifies a significant redesign focusing on the tent's interior to maximize the field of view for photography, specifically through designated openings. This change goes beyond merely adjusting the size, shape, weight, material, safety, durability, aesthetics, ergonomics, cost, setup, or transport of the tent. It addresses an innovative use of the tent interior for wildlife observation through photography that does not necessarily involve leaving the tent, which represents a unique insight into how tents can be utilized. This need clearly falls under the first criterion for being considered a latent need, as it is a significant change to the product design that does not fit into the specified exclusion categories. Furthermore, it reflects an exceptionally innovative and clearly expressed insight regarding the product and how it is used, making it a valid example of a latent need.	*LLM's chain of thought:* The customer's need is centered on enhancing the durability of the tent floor, specifically making it resistant to tears from sharp objects and general wear. This need falls directly into one of the predefined categories (durability) that does not qualify as a latent need according to the stated criteria. Therefore, despite being a valid concern and a potential area for product improvement, this need does not represent a significant deviation from existing product design paradigms or introduce an exceptionally innovative insight. Thus, it does not meet the conditions for being labeled as a latent need.

Note: The LLM correctly classifies both examples.

7 Cost Analysis

Cost is an important consideration in latent need identification studies, and the use of LLMs can be a cost-effective alternative. For example, *GPT-4-Turbo*, which was used in this study, costs 10 USD per 1 million input tokens and 30 USD per 1 million output tokens. This resulted in minimal costs for experiments 1 and 3, amounting to just a few cents, while experiment 2, which involved generating approximately 80,000 tokens, incurred a total cost of around 2.4 USD. These costs are significantly lower than those associated with traditional user studies and surveys, which often require substantial investments in participant recruitment, compensation, and data analysis. The cost-effectiveness of using LLMs for latent needs identification has important implications for the scalability and accessibility of this research, enabling researchers and organizations with limited budgets to conduct comprehensive studies and obtain valuable insights at a fraction of the cost of conventional methods.

8 Limitations

The limitations of utilizing LLMs in requirements elicitation deserve detailed examination. One significant limitation is the inherent bias present in LLMs. These models are trained on vast datasets that are predominantly sourced from the Internet, which tends to reflect the biases and inequalities of the societies that produced the data. As a result, LLMs can inadvertently perpetuate stereotypes and may disproportionately represent certain cultural and socioeconomic perspectives while underrepresenting others. For instance, Western-centric viewpoints are often overrepresented in LLMs that are trained on mostly English text, leading to a skewed understanding of global user needs and perspectives.

Another limitation lies in the uneven representation of certain cultures and personalities. LLMs are more likely to generate content that mirrors the dominant socioeconomic narratives found in their training data. This means that the needs and preferences of less-documented communities may not be adequately captured or prioritized. For example, the design requirements of users from low-income regions may be overlooked, resulting in products that do not fully address the diverse needs of the global population.

Moreover, the ability of LLMs to identify latent needs is constrained by their training data. While LLMs can simulate a broad range of user interactions and generate diverse user personas, they are ultimately limited by the scope and diversity of the data they have been trained on. If the training data lack representation of certain user experiences or contexts, the LLMs' output will similarly lack this diversity.

The process of creating user agents and simulating product experiences, as implemented in frameworks like Elicitron, can be automated and scaled. However, the quality and authenticity of these simulations depend heavily on the context provided and the diversity of the agents generated. Inadequate contextualization can lead to repetitive or generic outputs that fail to capture the varied needs of real-world users. Additionally, while techniques like chain-of-thought reasoning can enhance the interpretability and reasoning capabilities of LLMs, they do not fully mitigate the risk of missing out on unique or less obvious user needs that fall outside the patterns recognized by the model.

9 Conclusions

This article presents Elicitron, a framework that leverages LLMs to enhance requirements elicitation and uncover diverse user needs, including those that are latent. Our context-aware serial generation method proved most effective in creating diverse user agents. Elicitron successfully outperformed traditional empathic lead user interviews in generating latent needs. LLMs also demonstrate effectiveness in analyzing interviews and automatically classifying latent needs. Elicitron shows the potential of leveraging LLMs for requirements elicitation and user-centered design, providing an alternative and cost-effective method for designers.

While Elicitron shows promise, the quality of insights depends on the LLM's capabilities, and prioritizing latent needs remains a designer's task. Future work includes user studies to validate Elicitron's ability to aid designers and exploring multiagent interactions to uncover broader unmet needs. Moreover, exploring the possibility of incorporating multimodal inputs and outputs in the process of eliciting requirements represents another prospective avenue for research.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request.

References

[1] Zave, P., 1997, "Classification of Research Efforts in Requirements Engineering," ACM Comput. Surv. (CSUR), **29**(4), pp. 315–321.

[2] Berry, D. M., 2007, "Ambiguity in Natural Language Requirements Documents," Monterey Workshop, Monterey, CA, Sept. 10–13, Springer, pp. 1–7.

[3] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., and Neelakantan, A., et al. 2020, *Language Models Are Few-Shot Learners*," *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, eds., Curran Associates, Inc., New Orleans, LA, pp. 1877–1901.

[4] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I., 2017, "Attention Is All You Need," Advances in Neural Information Processing Systems, Vol. **30**, pp. 6000–6010.

[5] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, Ed., Le, Q. V., and Zhou, D., 2022, "Chain-of-thought Prompting Elicits Reasoning in Large Language Models," Adv. Neural Inf. Process. Syst., **35**, pp. 24824–24837.

[6] Maalej, W., Nayebi, M., Johann, T., and Ruhe, G., 2015, "Toward Data-Driven Requirements Engineering," IEEE Softw., **33**(1), pp. 48–54.

[7] Lin, J., and Seepersad, C. C., 2007, "Empathic Lead Users: The Effects of Extraordinary User Experiences on Customer Needs Analysis and Product Redesign," International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Las Vegas, NV, Sept. 4–7, pp. 289–296.

[8] Lingard, L., 2023, "Writing With ChatGPT: An Illustration of Its Capacity, Limitations & Implications for Academic Writers," Perspect. Med. Educ., **12**(1), p. 261.

[9] Htet, A., Liana, S. R., Aung, T., and Bhaumik, A., 2024, "ChatGPT in Content Creation: Techniques, Applications, and Ethical Implications," *Advanced Applications of Generative AI and Natural Language Processing Models*, A. J. Obaid, et al., eds., IGI Global, Hershey, PA, pp. 43–68.

[10] Subagja, A. D., Ausat, A. M. A., Sari, A. R., Wanof, M. I., and Suherlan, S., 2023, "Improving Customer Service Quality in MSMEs Through the Use of ChatGPT," J. Minfo Polgan, **12**(2), pp. 380–386.

[11] Li, Y., Choi, D., Chung, J., Kushman, N., Schrittwieser, J., Leblond, R., and Eccles, T., et al. 2022, "Competition-Level Code Generation With AlphaCode," Science, **378**(6624), pp. 1092–1097.

[12] Doris, A. C., Grandi, D., Tomich, R., Alam, M. F., Ataei, M., Cheong, H., and Ahmed, F., 2024, "DesignQA: A Multimodal Benchmark for Evaluating Large Language Models' Understanding of Engineering Documentation," arXiv preprint arXiv:2404.07917.

[13] Etesam, Y., Cheong, H., Ataei, M., and Jayaraman, P. K., 2024, "Deep Generative Model for Mechanical System Configuration Design," arXiv preprint arXiv: 2409.06016.

[14] Shanahan, M., McDonell, K., and Reynolds, L., 2023, "Role Play With Large Language Models," Nature, **623**(7987), pp. 493–498.

[15] Csepregi, L. M., 2021, "The Effect of Context-Aware LLM-Based NPC Conversations on Player Engagement in Role-Playing Video Games," Unpublished Manuscript.

[16] Zhu, A., Martin, L., Head, A., and Callison-Burch, C., 2023, "CALYPSO: LLMs as Dungeon Master's Assistants," Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, Salt Lake City, UT, Oct. 8–12, pp. 380–390.

[17] Gray, C., Yilmaz, S., McKilligan, S., Daly, S., Seifert, C., and Gonzalez, R., 2015, "Idea Generation Through Empathy: Reimagining the 'Cognitive Walkthrough,'" ASEE Annual Conference & Exposition, Seattle, WA, June 14–17.

[18] Schmitt, E., and Morkos, B., 2016, "Teaching Students Designer Empathy in Senior Capstone Design," Capstone Design Conference, Columbus, OH, June 6–8.

[19] Walther, J., Miller, S. E., and Kellam, N. N., 2012, "Exploring the Role of Empathy in Engineering Communication Through a Transdisciplinary Dialogue," 2012 ASEE Annual Conference & Exposition, San Antonio, TX, June 10–13, pp. 25–622.
[20] Hannukainen, P., and Holtta-Otto, K., 2006, "Identifying Customer Needs: Disabled Persons as Lead Users," International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Philadelphia, PA, Sept. 10–13, pp. 243–251.
[21] Leonard, D., and Rayport, J. F., 1997, "Spark Innovation Through Empathic Design," Harv. Bus. Rev., **75**, pp. 102–115.
[22] Otto, K. N., 2003, *Product Design: Techniques in Reverse Engineering and New Product Development*, Tsinghua University Press Co. Ltd, Hoboken, NJ.
[23] Ulrich, K. T., and Eppinger, S. D., 2016, *Product Design and Development*, McGraw-Hill, New York.
[24] Suh, N. P., 1990, *The Principles of Design*, Oxford University Press, New York.
[25] Von Hippel, E., 1986, "Lead Users: A Source of Novel Product Concepts," Manage. Sci., **32**(7), pp. 791–805.
[26] Urban, G. L., and Von Hippel, E., 1988, "Lead User Analyses for the Development of New Industrial Products," Manage. Sci., **34**(5), pp. 569–582.
[27] Issa, N. Md., Sasaki, H., Okamura, N., Yahya, W. J., Rahman, M. A. A., Ariff, M. H. M., and Koga, T., 2023, "Proposition and Verification of a Design Method to Discover Latent Needs Based on Empathy, Experiences, and Working Prototype by Designing Autonomous Childcare Vehicle," J. Adv. Vehicle Syst., **14**(1), pp. 19–34.
[28] Qiu, Y., and Jin, Y., 2023, "Document Understanding-Based Design Support: Application of Language Model for Design Knowledge Extraction," ASME J. Mech. Des., **145**(12), p. 121401.
[29] Chen, L., Jing, Q., Tsang, Y., Wang, Q., Sun, L., and Luo, Ji., 2024, "DesignFusion: Integrating Generative Models for Conceptual Design Enrichment," ASME J. Mech. Des., **146**(11), p. 111703.
[30] Zhu, Q., and Luo, J., 2023, "Toward Artificial Empathy for Human-Centered Design: A Framework."
[31] Zhu, Q., Chong, L., Yang, M., and Luo, J., 2024, "Reading Users' Minds From What They Say: An Investigation Into LLM-Based Empathic Mental Inference," arXiv preprint arXiv:2403.13301.
[32] Barandoni, S., Chiarello, F., Cascone, L., Marrale, E., and Puccio, S., 2024, "Automating Customer Needs Analysis: A Comparative Study of Large Language Models in the Travel Industry," arXiv preprint arXiv:2404.17975.
[33] Strobel, J., Hess, J., Pan, R., and Wachter Morris, C. A., 2013, "Empathy and Care Within Engineering: Qualitative Perspectives From Engineering Faculty and Practicing Engineers," Eng. Stud., **5**(2), pp. 137–159.
[34] Raviselvam, S., Sanaei, R., Blessing, L., Hölttä-Otto, K., and Wood, K. L., 2017, "Demographic Factors and Their Influence on Designer Creativity and Empathy Evoked Through User Extreme Conditions," International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Cleveland, OH, Aug. 6–9, American Society of Mechanical Engineers, p. V007T06A011.
[35] Surma-Aho, A., Björklund, T., and Hölttä-Otto, K., 2018, "An Analysis of Designer Empathy in the Early Phases of Design Projects," DS 91: Proceedings of NordDesign 2018, Aug. 14–17, Linköping, Sweden.
[36] Tang, X., 2018, "From 'Empathic Design' to 'Empathic Engineering': Toward a Genealogy of Empathy in Engineering Education," 2018 ASEE Annual Conference & Exposition, Salt Lake City, UT, June 23–27.
[37] Boden, M. A., 2009, "Computer Models of Creativity," AI Mag., **30**(3), pp. 23–23.
[38] Amabile, T. M., 1988, "A Model of Creativity and Innovation in Organizations," Res. Organ. Behav., **10**(1), pp. 123–167.
[39] Amabile, T. M., 1982, "Social Psychology of Creativity: A Consensual Assessment Technique," J. Personal. Soc. Psychol., **43**(5), p. 997.
[40] Miller, S. R, Hunter, S. T., Starkey, E., Ramachandran, S., Ahmed, F., and Fuge, M., 2021, "How Should We Measure Creativity in Engineering Design? A Comparison Between Social Science and Engineering Approaches," ASME J. Mech. Des., **143**(3), p. 031404.
[41] Amabile, T. M., 2018, *Creativity in Context: Update to the Social Psychology of Creativity*, Routledge, New York.
[42] Regenwetter, L., Srivastava, A., Gutfreund, D., and Ahmed, F., 2023, "Beyond Statistical Similarity: Rethinking Metrics for Deep Generative Models in Engineering Design," Comput. Aid. Des., **165**, p. 103609.
[43] Ma, K., Grandi, D., McComb, C., and Goucher-Lambert, K., 2023, "Conceptual Design Generation Using Large Language Models," International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Boston, MA, Aug. 20–23.
[44] Jiralerspong, M., Bose, J., Gemp, I., Qin, C., Bachrach, Y., and Gidel, G., 2024, "Feature Likelihood Score: Evaluating the Generalization of Generative Models Using Samples," *Advances in Neural Information Processing Systems*, Vol. 36.
[45] Sarica, S., and Luo, J., 2023, "Innovation Slowdown: Decelerating Concept Creation and Declining Originality in New Technological Concepts," arXiv preprint arXiv:2303.13300.
[46] Picard, C., Schiffmann, J., and Ahmed, F., 2023, "DATED: Guidelines for Creating Synthetic Datasets for Engineering Design Applications," arXiv preprint arXiv:2305.09018.
[47] Regenwetter, L., Abu Obaideh, Y., and Ahmed, F., 2023, "Counterfactuals for Design: A Model-agnostic Method for Design Recommendations," International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Boston, MA, Aug. 20–23, American Society of Mechanical Engineers, p. V03AT03A008.
[48] Bagazinski, N. J., and Ahmed, F., 2023, "ShipGen: A Diffusion Model for Parametric Ship Hull Generation With Multiple Objectives and Constraints," J. Marine Sci. Eng., **11**(12), pp. 2215.
[49] Fan, J., Vuaille, L., Bäck, T., and Wang, H., 2023, "On the Noise Scheduling for Generating Plausible Designs With Diffusion Models," arXiv preprint arXiv:2311.11207.
[50] Rousseeuw, P. J., 1987, "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis," J. Comput. Appl. Math., **20**, pp. 53–65.
[51] Podani, J., 2009, "Convex Hulls, Habitat Filtering, and Functional Diversity: Mathematical Elegance Versus Ecological Interpretability," Commun. Ecol., **10**(2), pp. 244–250.
[52] Mueller, C. T., and Ochsendorf, J. A., 2015, "Combining Structural Performance and Designer Preferences in Evolutionary Design Space Exploration," Autom. Construct., **52**, pp. 70–82.
[53] Brown, N. C., and Mueller, C. T., 2019, "Quantifying Diversity in Parametric Design: A Comparison of Possible Metrics," AI EDAM, **33**(1), pp. 40–53.
[54] Zanitti, M., Sørensen, J., Terolli, E., and Kosta, S., 2022, Exploiting Consumption Diversity and neighbour Similarity Trade-offs in Recommender Systems: a User-Centric Offline Evaluation of Diversity Objectives.
[55] Chaudhuri, A., Sarma, M., and Samanta, D., 2019, "Advanced Feature Identification Towards Research Article Recommendation: A Machine Learning Based Approach," TENCON 2019-2019 IEEE Region 10 Conference (TENCON), Kerala, India, Oct. 17–20, IEEE, pp. 7–12.
[56] Holbrook III, H., 1990, "A Scenario-Based Methodology for Conducting Requirements Elicitation," ACM SIGSOFT Softw. Eng. Notes, **15**(1), pp. 95–104.
[57] OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., and Almeida, D., 2024, "GPT-4 Technical Report," URL 2303.08774.
[58] Van der Maaten, L., and Hinton, G., 2008, "Visualizing Data Using t-SNE," J. Mach. Learn. Res., **9**(11), pp. 2579–2605.
[59] Hripcsak, G., and Rothschild, A. S., 2005, "Agreement, the f-Measure, and Reliability in Information Retrieval," J. Am. Med. Inf. Assoc., **12**(3), pp. 296–298.