npj | computational materials

Article

Published in partnership with the Shanghai Institute of Ceramics of the Chinese Academy of Sciences

https://doi.org/10.1038/s41524-024-01423-2

Opportunities for retrieval and tool augmented large language models in scientific facilities

Michael H. Prince[1], Henry Chan[2], Aikaterini Vriza[2], Tao Zhou[2], Varuni K. Sastry[3], Yanqi Luo[1], Matthew T. Dearing[4], Ross J. Harder[1], Rama K. Vasudevan[5] & Mathew J. Cherukara[1]

Upgrades to advanced scientific user facilities such as next-generation x-ray light sources, nanoscience centers, and neutron facilities are revolutionizing our understanding of materials across the spectrum of the physical sciences, from life sciences to microelectronics. However, these facility and instrument upgrades come with a significant increase in complexity. Driven by more exacting scientific needs, instruments and experiments become more intricate each year. This increased operational complexity makes it ever more challenging for domain scientists to design experiments that effectively leverage the capabilities of and operate on these advanced instruments. Large language models (LLMs) can perform complex information retrieval, assist in knowledge-intensive tasks across applications, and provide guidance on tool usage. Using x-ray light sources, leadership computing, and nanoscience centers as representative examples, we describe preliminary experiments with a Context-Aware Language Model for Science (CALMS) to assist scientists with instrument operations and complex experimentation. With the ability to retrieve relevant information from facility documentation, CALMS can answer simple questions on scientific capabilities and other operational procedures. With the ability to interface with software tools and experimental hardware, CALMS can conversationally operate scientific instruments. By making information more accessible and acting on user needs, LLMs could expand and diversify scientific facilities' users and accelerate scientific output.

Conversational agents such as ChatGPT built on large language models (LLMs) are upending various industries and professions[1]. Their impact spans education[2,3], talent acquisition[4], and many other sectors. A report by McKinsey and Company estimates that generative AI, including LLMs, could add trillions of dollars annually to worldwide productivity[5].

LLMs also have the potential to revolutionize every aspect of scientific discovery, including accelerating literature search, aiding in experimental design, providing data summaries, and publication writing and editing[6]. LLMs have also been shown to be fast learners, requiring only a few training examples to quickly gain domain expertise, e.g., for materials property prediction and inverse design[7].

At scientific user facilities, such agents could significantly contribute to nearly all facets of operating advanced scientific instruments and conducting experiments. These agents could help new users navigate different aspects of the facility, such as the proposal submission process, safety policies, and operating practices. They could also help users design suitable experiments to address their scientific inquiries, including selecting the appropriate instrument, determining the necessary modalities, and advising on the sample preparation. Additionally, during the experiment, they could guide basic instrument operation. Despite their potential, LLMs suffer from several dangerous limitations, including the tendency to 'hallucinate' answers when faced with questions not part of their training.

We describe preliminary experiments with retrieval[8] and tool-augmented[9] LLMs tailored for scientists' needs. Scientific context-aware LLMs leverage existing data stores to augment their capabilities, allowing them to perform better in domains that are not part of the pre-training data, thereby eliminating the need to perform any fine-tuning on the domain-specific information that can be computationally expensive and time-

[1]Advanced Photon Source, Argonne National Laboratory, Lemont, IL, USA. [2]Center for Nanoscale Materials, Argonne National Laboratory, Lemont, IL, USA. [3]Argonne Leadership Computing Facility, Argonne National Laboratory, Lemont, IL, USA. [4]Business and Information Systems, Argonne National Laboratory, Lemont, IL, USA. [5]Center for Nanophase Materials, Oak Ridge National Laboratory, Oak Ridge, TN, USA. e-mail: mcherukara@anl.gov

consuming. In this work, we describe using LLMs to create an agent with knowledge of scientific user facilities. We describe the initial results of assisting scientists through basic experiment planning assistance, instrument operation, and direct experimental tool control.

LLMs have demonstrated strong performance as autonomous agents across multiple science domains[10]. The key components that make the LLM function as the agent's brain are memory, planning, and tool use. Recently, Yager discussed the importance of memory by building domain-specific chatbots for physical science research. Their demonstration involves the creation of a context window using text chunks from scientific documents in PDF format. When the LLM is composing its reply, it uses this context to formulate the answer[11].

ChemCrow is another example of a domain-specific LLM agent designed to accomplish tasks related to materials design, organic synthesis, and drug discovery by integrating several chemistry tools. Their workflow combines planning via Chain of Thought reasoning (CoT) and tool utilization[12]. In another recent work, researchers built an Intelligent Agent based on GPT, capable of searching hardware documentation and controlling multi-instrument systems to conduct experiments. An important consideration in this type of research is the evaluation of the chatbots or LLMs for scientific applications. In this context, Zaki et al. developed a question-answering dataset for measuring the performance of GPT-3.5 and GPT-4. However, the performance is measured without considering the improvement when an LLM is provided with additional domain-specific scientific context[13]. Jablonka et al. described results from a hackathon applying LLMs to various scientific tasks[14].

Fig. 1 | Overview of CALMS: CALMS uses a large language model in conjunction with conversational memory, document stores, and experimental tools to answer user queries or take action to drive an instrument.

Results

Overview of CALMS

The core of CALMS consists of four components: a large language model (LLM)[15], conversational history that allows follow-on queries, semantic search over document stores to retrieve the most relevant context given a question from the user[16], and instrument tools that the LLM can use when instructed to by the user[17]. Figure 1 shows a schematic of the structure of CALMS. When provided with appropriate context and tools, CALMs can either answer complex technical questions that the LLM has no prior knowledge of from its training[18] or execute an experiment or perform a computation using the appropriate tools[17]. We describe each component in detail in the following sections. We compare responses from two state-of-the-art LLMs, OpenAI's GPT-3.5 Turbo and an open-source model Vicuna, over questions related to experimental planning assistance, operation, and ability to drive an instrument successfully. We note that the CALMS framework is independent of the LLM, and other open-source or closed-source LLMs can be swapped without changing the framework.

GPT-3.5

The Generative Pretrained Transformer 3.5 (GPT-3.5) is a near state-of-the-art large language model developed by OpenAI and released in late 2022. Similar to its predecessors and successor, GPT-4, the model architecture is based on the Transformer[19], which has demonstrated effectiveness in various natural language processing tasks, including generating text, answering questions, writing content with human-like creativity, and even generating computer programming code. These outcomes result from predictive inference based on an input sequence of text processed through the language patterns the model learned from a vast training set.

The most performant of the *pre*-GPT-4 class of models, GPT-3.5 Turbo, was trained with 175 billion parameters and is available at the time of this writing with context window sizes of 4096 or 16,384 tokens. The context window corresponds to the maximum number of tokens (i.e., words, punctuation, or even sub-strings of words) the model can process during a single inference procedure. The content included in this window represents a simple type of memory that can be leveraged to provide background as effective guidance toward predicting subsequent words. This feature can significantly impact the quality and relevancy of the model output and has led to the burgeoning field of "prompt engineering" to develop and implement best practices.

While the model is proprietary, Microsoft makes it available through the Azure cloud with an enterprise subscription. Hosting of the model resides on a dedicated and restricted instance, so data remains local, is not retained by default, and is not shared with third parties. This approach alleviates some challenges with using public models, such as ChatGPT from OpenAI.

The interaction with GPT-3.5 starts with an API call authenticated using a unique key. This key is essential for accessing OpenAI's capabilities securely and ensuring that only authorized users can make requests. Before making the API call, specific parameters, i.e., temperature and top_p, guide the model's response generation. The temperature parameter controls the randomness of the output and takes values between 0-1. A higher temperature leads to more creative output, while a lower temperature results in more deterministic outputs. The top_k parameter limits the model's choices to the 'k' most likely next words at each step of the generation process. The top_p parameter is an alternative to the temperature that focuses on a subset of predictions that cumulatively add up to the probability 'p'. After these parameters are set and the API call is made, the model can be used either for chat completion or embeddings generation. For chat completion tasks, the model generates a continuation of the provided text input based on the context and parameter set. For the embeddings generation task, the model processes the input text to create a vector representation that captures the semantic meaning of the text. These embeddings can be used for various applications like text similarity assessment or semantic search.

Open-source LLMs

LLMs have transformed the AI field in recent years with the advent of large and complex models like GPT-3 and GPT-4. Though their performance and capabilities are unparalleled, they often require very large compute resources, thus limiting the number of individuals and enterprises that can develop and maintain systems of this scale. Moreover, with limited or no transparency about the model specifications, data, and development methods, these "closed" source LLMs raise concerns regarding adherence to AI ethical standards (including fairness, morality, accountability, and misuse, among others). Open-source LLMs alleviate these concerns by

Table 1 | A comparison of the benchmarking results for the two models experimented with in this work

	Vicuna	GPT 3.5
MMLU	56.67%	70.0%
HellaSwag	81.24%	85.5%
ARC Challenge	57.08%	85.2%
WinoGrande	74.66%	81.6%
GSM-8K	11.3%	57.1%

Taken from https://arxiv.org/pdf/2303.08774v5.pdf and https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

democratizing the AI community with free and unrestricted accessibility to models and information driven by platforms and communities with a transparent and collaborative approach. Open-source LLMs also provide the advantage of being accessible to modification to tailor it to a specific task. HuggingFace is such a resource that provides a platform where researchers, academics, and data scientists can share datasets[20]. HuggingFace also provides the necessary tools within their Transformers library to fine-tune and deploy ML workflows for open-source models, including LLMs. With a wide range of modalities of models and data available, it also has the added advantage of seamlessly switching and testing between different models with minimal code changes.

HuggingFace also provides the "Open LLM Leaderboard evaluation suite" that evaluates and compares the performance of different LLMs based on a relative Elo rating strategy. Traditionally, the Elo rating system has been a method used in the field of chess to calculate the relative skill levels of players based on the outcome of the games. A similar strategy is applied to the model responses, where human annotators rate responses for helpfulness and truthfulness in a pairwise setting[21]. At the time of finalizing the paper, "lmsys/vicuna-13b-v1.5" was ranked among the top five pertained models with an average score of 61.63. This model was obtained by fine-tuning Llama 2 on supervised instruction downstream tasks trained on conversations collected from ShareGPT with a sequence length of 2048[22]. For a more detailed comparison, various benchmarks results compiled from OpenAI and the Open LLM Leaderboard are presented in Table 1.

In the following sections, we compare responses from Vicuna 1.5-16k against GPT-3.5 across a series of tasks. Our evaluation encompasses tasks including general comprehension of user questions, domain-specific context retrieval, quality of the answers, and utilization of tools.

Context retrieval through semantic search

One of the prominent use cases of LLMs is context retrieval through semantic search, which is a process that involves extracting relevant information from a large text body based on the meaning and context of the query instead of relying on keyword matching. Currently, Claude has the largest context window of 100k tokens, corresponding to around 75,000 words.

Two approaches are possible to extend the context retrieval capabilities of models beyond their training corpus. First, fine-tuning the model with domain-specific training data requires the careful curation of data and often necessitates significant up-front investment either in local GPU resources (e.g., Llama was trained on 2048 GPUs)[23] in the case of open-source or in cloud compute costs in the case of closed-source models. A great deal of expertise in language modeling, distributed training, transformers, and data curation is also typically required to fine-tune these models successfully. However, this approach will likely provide the best results when a sufficiently large corpus of data for fine-tuning is available.

A second approach to enhance the capabilities of LLMs is to leave the model weights unchanged but provide appropriate additional context to help answer user queries. This strategy is typically cost-efficient, and a plethora of frameworks now exist to enable facile interaction of the LLM with document stores. A remaining challenge, however, is that even the largest models, e.g., GPT-4-128k, can only accept input text up to 128k tokens, corresponding to approximately 300 pages of text. While future models might allow even larger input sizes, the current limitations prevent passing in all relevant documentation[24].

To address this gap, available documentation is pre-processed for context retrieval, which is performed via a lightweight embedding model. This process known as retrieval-augmented generation (RAG) is being widely used to enhance the domain knowledge of LLMs[25]. RAG models have shown significant improvements in various natural language processing tasks by leveraging external knowledge effectively. They are particularly useful in scenarios where the required information for generating a response is not contained within the input query alone, making them a powerful tool for enhancing AI's understanding and generation capabilities. For tests with OpenAI models, the OpenAI embedding model was used to embed and retrieve results. The open-source tests used the "all-mpnet-base-v2" model. All technical documents are split into token windows of a pre-determined size, and their embeddings are stored in a vector database, e.g., ChromaDB, collection[26]. When a question is sent, the user's text is embedded, and the top $N = 4$ text chunks closest to the user input are retrieved from the store. These context chunks are then included in the prompt provided to the conversational LLM.

Conversational memory

At their core, LLMs are probabilistic models that learn sentence or paragraph completion. In other words, given a sequence of words, they are trained to predict the most likely completions of that sentence or paragraph. Models that have been fine-tuned for chat have an additional training step that teaches them how to complete conversations, either through an unsupervised approach by learning from human-provided conversations or, more popularly, through reinforcement learning with human feedback (RLHF)[27]. Such fine-tuned chat models can complete human-like conversations, and we leverage this ability by providing the conversational history as part of the subsequent prompt. As previously discussed, due to the limited amount of text that can be provided in the prompt, reproducing the entire conversational history is often not possible. In essence, the problem is how to provide the LLM with short-term memory to answer questions more effectively when faced with a limited context window. Several strategies have been proposed to address this; one option that CALMS adopted is to use a moving window over the conversation history to include only the last $K = 6$ conversations between the user and the LLM. Supplementary Fig. 1A shows the prompt to the LLM with context and memory.

Software and experimental tools

To further augment the use of LLMs, current research is exploring if the LLMs can go beyond text and reason about the physical world. For example, work from Google and Microsoft have demonstrated using NLP to achieve robotic tasks[28,29]. This process involves building a series of design principles that include high-level robot APIs or function libraries, special prompting structures, and human feedback via text.

Building upon this strategy, the current implementation of CALMS incorporates capabilities such as calling the Materials Project API[30], an instrument control software (spec™)[31], and is based on Chain-of-Thought prompting and the ReAct framework[32,33]. These functions are provided as a list of tool names, including the description of the way they interact with scientific equipment and details about the user inputs and outputs. The tool is then instructed to answer a user-provided prompt by leveraging the available tools. A parser interprets the LLM output for tool call requests. If a tool is called, then the parser inserts the output into the prompt, and the model continues generating actions until a response is provided. Supplementary Fig. 1B shows the prompt to the LLM with the tool description.

In effect, using conversational memory, context-retrieval, and specified tools creates a dynamic pre-prompt that allows the LLM to parse and respond to user input with a detailed understanding of the context behind those queries.

		Experimental Planning Assistance											
		Without context						With context					
		Vicuna			GPT-3.5			Vicuna			GPT-3.5		
Facility	Example	Rel	Hal	Com	Rel	Hal	Com	Rel	Hal	Com	Rel	Hal	Com
CNM	E1	Yes	Yes	0	Yes	Yes	0	Yes	No	4	Yes	No	5
ALCF	E2	N/A	No	0	No	No	0	Yes	No	2	Yes	No	5
APS	E3	Yes	Yes	0	Yes	Yes	3	Yes	Yes	1	Yes	No	4
CNMS	E4	No	Yes	0	No	Yes	0	Yes	No	4	Yes	No	5
APS	E5	Yes	No	0	Yes	No	2	Yes	No	5	Yes	No	5
APS	E6	Yes	Yes	3	Yes	Yes	2	Yes	No	5	Yes	No	5
APS	E7	N/A	No	0	Yes	No	4	Yes	No	4	Yes	No	5
APS	E8	No	No	0	No	No	0	N/A	No	N/A	Yes	No	5
APS	E9	No	Yes	0	No	No	0	N/A	No	N/A	Yes	No	5
ALCF	E10	No	Yes	0	No	Yes	0	Yes	No	4	Yes	No	5
ALCF	E11	No	Yes	0	No	Yes	0	Yes	No	5	Yes	No	5
ALCF	E12	No	Yes	0	No	Yes	0	Yes	No	4	Yes	No	5
CNM	E13	N/A	No	0	N/A	Yes	0	Yes	No	4	Yes	No	5
CNM	E14	No	Yes	0	No	Yes	0	Yes	No	2	Yes	No	5
		4 / 11	5 / 14	0.2	4 / 12	5 / 14	0.8	12 / 12	13 / 14	3.7	14 / 14	14 / 14	4.9

Notes:

E2: Without context, both models stated not enough information and asked for more context from the user. No hallucination.

E7: Vicuna without context stated not enough information. No hallucination.

E8: Vicuna without context stated not enough information and no hallucination. GPT-3.5 without context stated 2 real examples and hallucinated 1 but also asked for more context.

E9: Vicuna without context stated a true, but incorrect concept. No hallucination. GPT 3.5 without context stated a true, but incorrect concept.

E13: Vicuna without context: stated that the term can refer to different things depending on the context. GPT 3.5 without context has a mild hallucination in "polymer robot".

Situation	Response Rel	Response Hal
Declines to answer, assumes nothing.	N/A	No
Declines to answer, inserts real information.	No	No
Declines to answer, hallucinates.	No	Yes
Answers, hallucinates.	No	Yes

Fig. 2 | Scoring of experimental planning assistance questions with and without context. We score the models on relevance, absence of hallucination, and completeness of response.

AI-assisted experimental planning assistance

We first explore the ability of context-aware LLMs, such as those implemented in CALMS, to respond to highly technical questions typical of those encountered by new scientific facility users. CALMS is queried with questions concerning user facilities, including the Advanced Photon Source (APS), Center for Nanoscale Materials (CNM), Argonne Leadership Computing Facility (ALCF), and Center for Nanophase Materials Sciences (CNMS). A summary of responses with Vicuna and GPT-3.5 Turbo with and without context retrieval is shown in Fig. 2. The AI response is first graded by their helpfulness (denoted Rel in Fig. 2 for relevance), 'yes' for giving a relevant answer to the question, and 'no' for an irrelevant answer. Next, their truthfulness is evaluated by checking if hallucinations (denoted Hal) appear in the response. While we appreciate that relevance and hallucinations are not correlated, a relevant response can be hallucinated, and an irrelevant response can be entirely truth-based. Finally, the completeness (denoted Com) of the AI response is graded with a score of 0–5. Details on the scoring are included in the Methods section. A complete list of the queries and AI responses is found in the data availability section, with highlighted key differences. We caution that the reader might observe slightly different results because the LLMs outputs are non-deterministic.

Given an accurate context, CALMS can always provide relevant answers to questions specific to each user facility. Rarely was hallucination observed. In general, GPT-3.5 Turbo tends to provide a more complete answer than Vicuna, scoring multiple 5 s in the completeness grading. A score of 5 indicates that the users are provided with all the necessary information to select the appropriate tool to complete their task.

Without appropriate context, the language models have a strong tendency to hallucinate. As a representative example, we queried the models on the name of the automated image simulation framework developed at the CNM. Both Vicuna and GPT-3.5 Turbo provided the user with a made-up tool called ImageSim (an apparent portmanteau for image and simulation) (see Fig. 3). We note that both models, with context, were able to retrieve the correct answer, which is a tool called ingrained[34]. Without context, irrelevant answers have been observed, and occasionally, the models decline to answer the question, citing insufficient knowledge (see for example Supplementary Fig. 2). Supplementary Figs. 3–14 show further examples illustrating the importance of context-aware responses.

AI-assisted experiments

We further queried the models' ability to provide operational assistance on advanced scientific and computational resources at the abovementioned user facilities. A summary of responses with Vicuna and GPT-3.5 Turbo with and without context retrieval is shown in Fig. 4. Similar behavior was observed compared to what was noted for the experimental planning assistance queries. Without context, both Vicuna and GPT-3.5 Turbo tend to hallucinate. Even when they are being truthful (i.e., not hallucinating), the answers they provide are typically not directly helpful.

For instance, when queried about the procedure to start a tomographic scan, Vicuna, without context, responded with a truthful answer concerning the computational tomography (CT) scan in a hospital, which is unrelated to the tomographic scan performed at the APS (Fig. 5). In this case, the truthful yet unhelpful answer Vicuna provided was due to the LLM mistaking the tool being queried with a different subject well-known in the public domain. GTP-3.5 Turbo realized the nuances in the question and suggested the user be more specific about

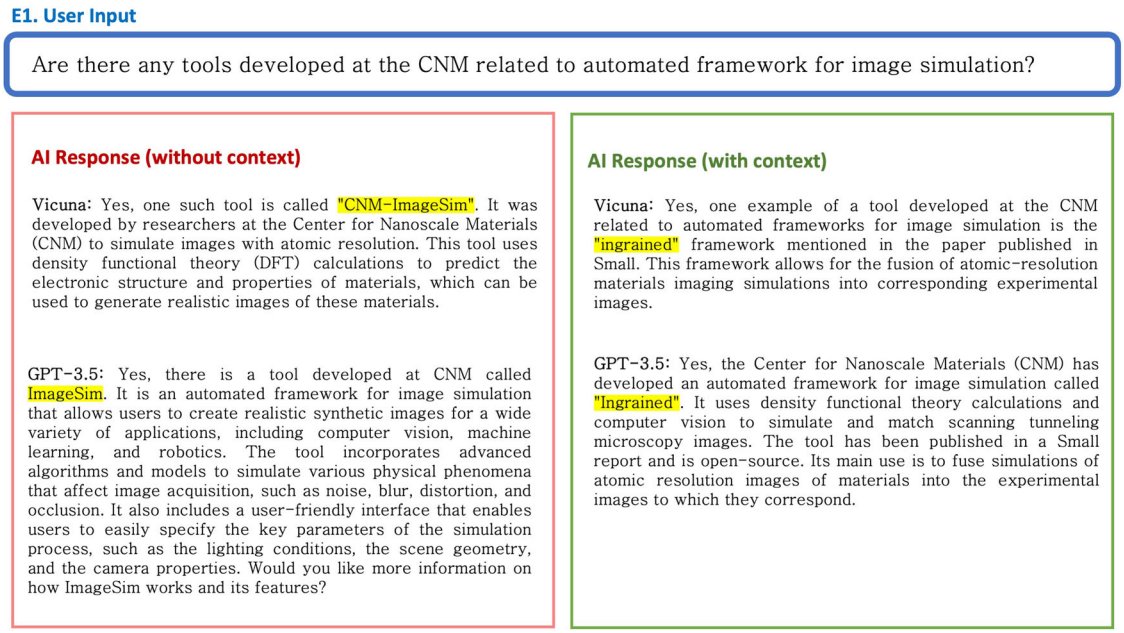

Fig. 3 | Demonstration of AI-assisted experimental planning assistance specific to the CNM user facility: (top) initial question from a user, (bottom-left) response from AI without context, (bottom-right) response from AI with context. Major differences between the responses are highlighted (in yellow) to illustrate the importance of context.

		Operational Assistance											
		Without context						With context					
		Vicuna			GPT-3.5			Vicuna			GPT-3.5		
Facility	Example	Rel	Hal	Com	Rel	Hal	Com	Rel	Hal	Com	Rel	Hal	Com
APS	O1	No	No	0	Yes	No	1	Yes	No	3	Yes	No	5
CNM	O2	Yes	Yes	0	Yes	Yes	0	Yes	No	4	Yes	No	3
ALCF	O3	No	Yes	0	No	Yes	0	Yes	No	3	Yes	No	5
ALCF	O4	No	Yes	0	No	Yes	0	Yes	No	5	Yes	No	4
ALCF	O5	No	Yes	0	No	Yes	0	Yes	No	5	No	Yes	0
CNMS	O6	No	Yes	0	No	Yes	0	Yes	Yes	3	Yes	No	5
APS	O7	No	No	2	Yes	No	3	No	Yes	2	Yes	No	5
CNM	O8	Yes	Yes	0	Yes	Yes	0	Yes	No	5	Yes	No	5
CNM	O9	No	Yes	0	No	Yes	0	Yes	No	2	Yes	No	2
CNM	O10	Yes	Yes	0	Yes	Yes	0	Yes	Yes	0	Yes	No	5
ALCF	O11	No	Yes	0	No	Yes	0	Yes	No	5	Yes	No	5
APS	O12	Yes	Yes	0	No	No	0	Yes	No	0	Yes	No	5
APS	O13	Yes	Yes	0	No	No	0	Yes	No	1	Yes	No	5
APS	O14	No	No	0	No	No	0	Yes	No	2	Yes	No	5
		5 / 14	3 / 14	0.1	2 / 11	5 / 14	0.3	13 / 14	11 / 14	2.9	13 / 14	13 / 14	4.2

Fig. 4 | Scoring of experimental operation assistance questions with and without context. As before, we score the models on relevance, absence of hallucination, and completeness of response.

the type of tomographic scan they are inquiring about. It then provided a general answer about the need to submit a user proposal to perform a tomography experiment.

However, with context, both Vicuna and GPT-3.5 Turbo provide consistently relevant and truthful answers, with GTP-3.5 Turbo again faring better on the score of completeness than Vicuna. Still using the tomographic scan procedure query as an example, Vicuna responded with a short answer regarding the use of the tomoscan command, while GTP-3.5 Turbo's answer included solutions with the graphic user and the command line interface with a reference to the relevant documentation (Fig. 5). Supplementary Figs. 15–26 show further instances of the importance of context-aware responses.

Automated execution

Through the ReAct framework[33] and API calls, the CALMS framework could be configured to drive experiments in an agentic manner. To demonstrate this capability, a test was conducted on a real-world diffractometer simulating a common user operation.

For a user to take a diffraction measurement, an area detector, usually mounted to a robotic arm, must be moved into a set position relative to the sample and incoming x-ray beam. This angle is determined by the intensity of the beam, the material being observed, and the peak position selected by the user. To do this process manually, the user must first look up lattice information via a materials database. Then, the user must copy over the lattice information and the selected peak into the beamline interface, controlled by SPEC, to move the motors into position.

To automate this process, two LLM calls were implemented: GetLatticeConstants and SetDiffractometmer. For the LLM to successfully perform the action, it must first input the material into a call that queries the Materials Project API to return the lattice constants to the model. Next, it must combine the output of this call with the peak positions from the user input to execute a second call. Figure 6 describes in detail the sequence and

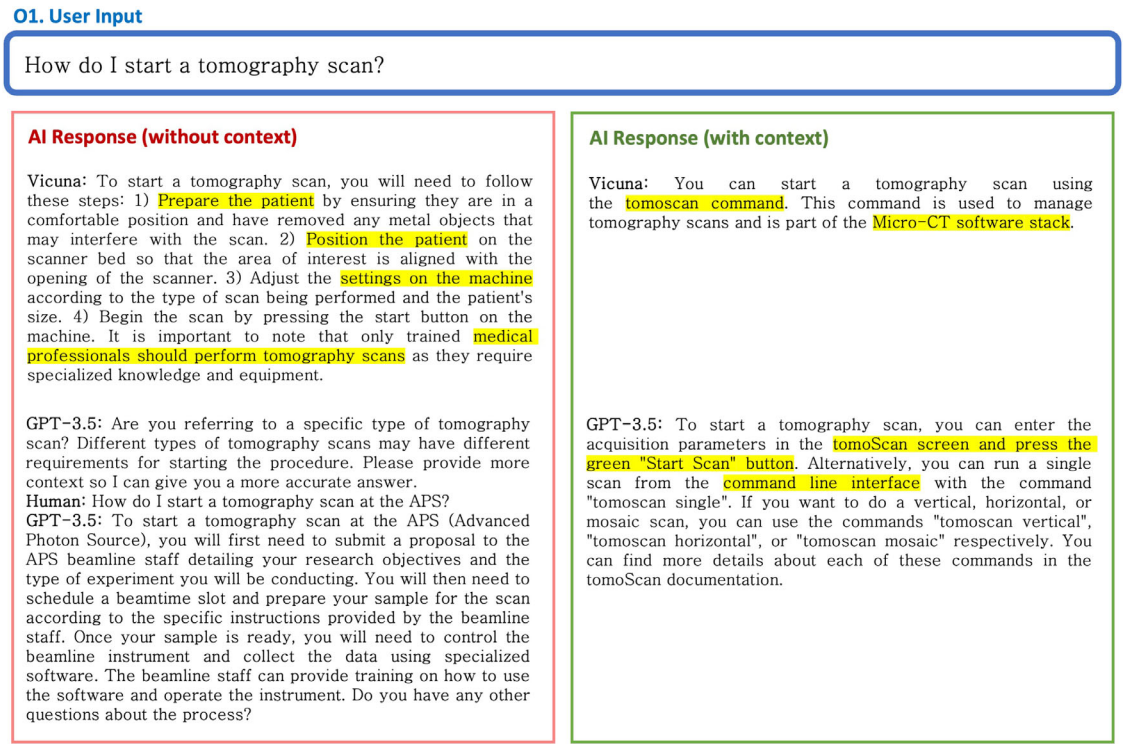

Fig. 5 | Demonstration of AI-powered operational assistance specific to the APS user facility: (top) initial query from a user, (bottom-left) response from AI without context, (bottom-right) response from AI with context. Major differences between the responses are highlighted (in yellow) to illustrate the importance of context.

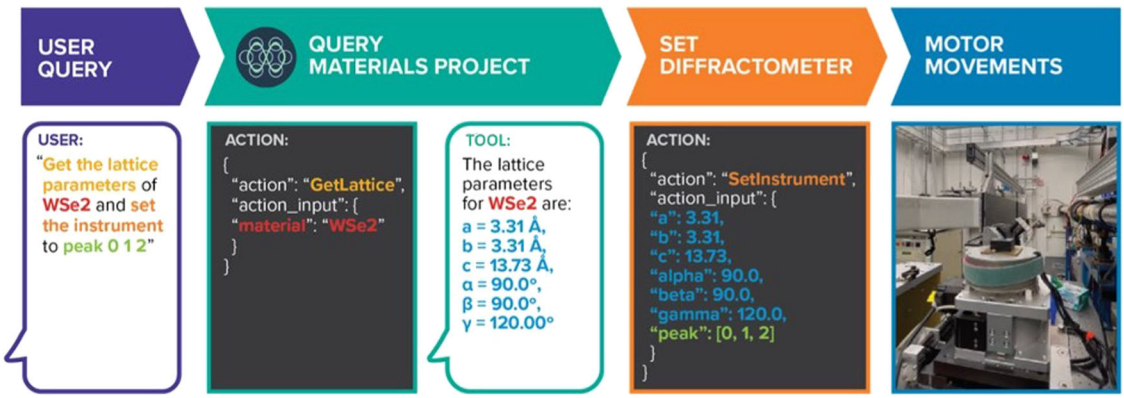

Fig. 6 | Workflow demonstrating the execution of a robot motor move based on a user query to CALMS. CALMS parses the user query to extract the material and Bragg peak, queries Materials Project for the lattice constants, and then calculates the position of and moves the beamline diffractometer to the Bragg peak requested by the user. A video of the demonstration is available at https://danielzt12.github.io/latest_news/2023/11/20/operating-scientific-instruments-with-LLMs.html.

parameters of the calls. In the example shown, a user requests the diffractometer to be set to the 012 Bragg peak of WSe_2. Within the CALMS framework, the LLM parses the query and recognizes the material and Bragg peak. It then uses a tool to call The Materials Project[30] to obtain the lattice information for WSe2. It then passes that information to a second tool that uses the Spec software to calculate the motor positions and then moves the diffractometer to that position.

These calls were implemented via LangChain's "Structured input ReAct" framework. This library API allows developers to define the parameters and types for multi-parameter input and requires the model to use valid JSON to perform a call. In our tests, we observed that the open-source models could not consistently follow this syntax. All results shown were run with OpenAI's GPT-3.5 model. We could execute this sequence of commands via a single user prompt on beamline 34 at the APS. A video of the commands being run on the beamline is provided in the supplemental material of this paper and can also be viewed online[35].

Discussion

In summary, we described the development of a Context-Aware Language Model for Science (CALMS) and exemplified the promising potential of leveraging LLMs for accelerating scientific experiments and discovery. Preliminary results obtained using CALMS backed by GPT-3.5 and Vicuna illustrate the applicability of LLMs in experimental planning assistance and operation guidance, leveraging the zero-shot capability of LLMs via prompt engineering[36]. The results emphasize the importance of context in addressing the typical issue of hallucination in LLMs. CALMS powered by GPT-3.5 and Vicuna both performed reasonably well in the Q&A tasks, but the results reveal limitations in the open-source models compared to state-of-

the-art closed-source models such as GPT-3.5 Turbo. This gap is particularly pronounced in tool usage, where we could not execute the experimental workflow with the open-source model Vicuna. With the investment and extraordinary rate of progress in open-source models and fine-tuning methods, we expect the performance gap to the closed-source models to narrow.

Traditionally, model performance over the years has shown power law scaling with an increase in the number of model parameters, dataset size, and compute budget. However, emerging research demonstrates that it is equally important to scale the training size with an increase in model parameter size to obtain optimal training[37]. Additionally, with instruction fine-tuning and RLHF, we see that the even smaller models of the size 1.3B parameters can equal the performance of the 175B GPT-3-base model[38]. With several innovations like SELF-Instruct[39] (that can enhance the model performance by 33% over the GPT3 base model with fine tuning on the self-generated instruction without any human intervention) and Chain-of-Thought prompting[15,40], the broad consensus is that these performance gains will continue to push the state of the art.

With such anticipated advanced model performance, we envision future capabilities could include leveraging decades of information recorded in e-logs as well as the ability to extract code or commands and execute experimental workflows fully autonomously. We envision context and tool aware language models such as CALMS to streamline the transfer of knowledge between instrument experts and domain users. The time saved from reduced user training and LLM assisted research planning provides time for thinking and exploring new science opportunities. This could enable a paradigm shift in the way users interact with the beamline instruments, paving the path to full automation and handling of exceptions based on past experience recorded in form of e-logs.

Methods

Scoring methodology

1/1 pt is automatically given if the response is rated 'yes' on relevance and 'no' on hallucination. 2/2 pts is given if the instructions can be executed by the user without error, 1/2 if minor errors are present that are easily fixable, and 0/2 if major errors are present in the response. 2/2 pts is assigned if the user is given all the information they need for their request, 1/2 if minor details are missing, and 0/2 if crucial details are missing that affect the execution. The maximum score is 5 pts.

Data availability

All data is available on GitHub (https://github.com/mcherukara/APS_ChatBot) and a snapshot at the time of writing is available at Zenodo (https://doi.org/10.5281/zenodo.11061583)

Code availability

All code and models are bundled with the data and available on GitHub (https://github.com/mcherukara/APS_ChatBot) and a snapshot at the time of writing is available at Zenodo (https://doi.org/10.5281/zenodo.11061583)

Received: 5 December 2023; Accepted: 16 September 2024;
Published online: 05 November 2024

References

1. Prepare for truly useful large language models. *Nat. Biomed. Eng.* **7**, 85–86. https://doi.org/10.1038/s41551-023-01012-6 (2023).
2. Wired. *Chegg Embraced AI. ChatGPT Ate Its Lunch Anyway*, https://www.wired.com/story/chegg-embraced-ai-chatgpt-ate-its-lunch-anyway/ (2023).
3. Kasneci, E. et al. ChatGPT for good? On opportunities and challenges of large language models for education. *Learn. Individual Differ.* **103**. https://doi.org/10.1016/j.lindif.2023.102274 (2023).
4. Clavié, B., Ciceu, A., Naylor, F., Soulié, G. & Brightwell, T. Large language models in the workplace: A case study on prompt engineering for job type classification. In *International Conference on Applications of Natural Language to Information Systems* 3–17 (2023).
5. Company, M. *The economic potential of generative AI: The next productivity frontier*, https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/the-economic-potential-of-generative-ai-the-next-productivity-frontier (2023).
6. Conroy, G. Scientists used ChatGPT to generate an entire paper from scratch—but is it any good? *Nature* **619**, 443–444 (2023).
7. Jablonka, K. M., Schwaller, P., Ortega-Guerrero, A. & Smit, B. Is GPT-3 all you need for low-data discovery in chemistry? *ChemRxiv preprint*. https://doi.org/10.26434/chemrxiv-2023-fw8n4 (2023).
8. Lewis, P. et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Adv. Neural Inf. Process. Syst.* **33**, 9459–9474 (2020).
9. Parisi, A., Zhao, Y. & Fiedel, N. Talm: Tool augmented language models. *arXiv preprint arXiv:2205.12255* (2022).
10. White, A.D. The future of chemistry is language. *Nat Rev Chem.* **7**, 457–458 (2023).
11. Yager, K. G. Domain-specific ChatBots for Science using Embeddings. *Digit. Discov.* **2**, 1850–1861 (2023).
12. Bran, M. et al. Augmenting large language models with chemistry tools. *Nat Mach Intell.* **6**, 525–535 (2024).
13. Zaki, M. & Krishnan, N. MaScQA: A Question Answering Dataset for Investigating Materials Science Knowledge of Large Language Models. *Digit. Discov*. **3**, 313–327 (2023).
14. Jablonka, K. M. et al. 14 examples of how LLMs can transform materials science and chemistry: a reflection on a large language model hackathon. *Digit. Discov.* **2**, 1233–1250 (2023).
15. Chiang, W.-L. et al. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. (2023).
16. Reimers, N. & Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proc. 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing,* 3982–3992 (2019).
17. Schick, T. et al. Toolformer: Language models can teach themselves to use tools. *Adv. Neural Inform. Process. Syst.* **36**, 68539–68551 (2024).
18. He, H., Zhang, H. & Roth, D. Rethinking with retrieval: Faithful large language model inference. *arXiv preprint arXiv:2301.00303* (2022).
19. Vaswani, A. et al. Attention is all you need. *Adv. Neural Inform. Process. Syst.* **30**, 6000–6010 (2017).
20. *HuggingFace*, https://huggingface.co/ (2023).
21. HuggingFace. *Can foundation models label data like humans?* https://huggingface.co/blog/llm-leaderboard (2023).
22. Zheng, L. et al. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *Neural Inform. Process. Syst.* **36**, 46595–46623 (2023).
23. Touvron, H. et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).
24. OpenAI. *New models and developer products announced at DevDay*, https://openai.com/blog/new-models-and-developer-products-announced-at-devday (2023).
25. Gao, Y. et al. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997* (2023).
26. Chroma. *Chroma*, https://docs.trychroma.com/ (2023).
27. Ziegler, D. M. et al. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593* (2019).
28. Driess, D. et al. Palm-e: an embodied multimodal language model. In *Proc. 40th International Conference on Machine Learning* in *Proceedings of Machine Learning Research,* Vol. 202, 8469–8488 (2023).
29. Vemprala, S., Bonatti, R., Bucker, A. & Kapoor, A. Chatgpt for robotics: design principles and model abilities. *Microsoft Auton. Syst. Robot. Res* **2**, 20 (2023).

30. Jain, A. et al. Commentary: The Materials Project: A materials genome approach to accelerating materials innovation. *APL Mater.* **1** (2013).
31. Software, C. S. *spec Software for Diffraction*, https://www.certif.com/content/spec/.
32. Wei, J. et al. Chain-of-thought prompting elicits reasoning in large language models. *Adv. Neural Inf. Process. Syst.* **35**, 24824–24837 (2022).
33. Yao, S. et al. React: Synergizing reasoning and acting in language models. *The International Conference on Learning Representations,* **11** (2023).
34. Schwenker, E. et al. Ingrained: an automated framework for fusing atomic-scale image simulations into experiments. *Small* **18**, 2102960 (2022).
35. Zhou, T. *Augmenting Scientific Instrumentation with LLMs*, https://danielzt12.github.io/latest_news/2023/11/20/operating-scientific-instruments-with-LLMs.html (2023).
36. Kojima, T., Gu, S. S., Reid, M., Matsuo, Y. & Iwasawa, Y. Large language models are zero-shot reasoners. *Adv. Neural Inf. Process. Syst.* **35**, 22199–22213 (2022).
37. Hoffmann, J. et al. Training compute-optimal large language models. In *Proc. International Conference on Neural Information Processing Systems,* Vol. 36, 30016–30030 (2022).
38. Ouyang, L. et al. Training language models to follow instructions with human feedback. *Adv. Neural Inf. Process. Syst.* **35**, 27730–27744 (2022).
39. Wang, Y. et al. Self-instruct: Aligning language model with self generated instructions. In *Proc. Annual Meeting of the Association for Computational Linguistics*, Vol. 61, 13484–13508 (2023).
40. Huang, J. et al. Large language models can self-improve. In *Proc. 2023 Conference on Empirical Methods in Natural Language Processing* 1051–1068 (2023).

Acknowledgements

Work performed at the Center for Nanoscale Materials and Advanced Photon Source, both U.S. Department of Energy Office of Science User Facilities, was supported by the U.S. DOE, Office of Basic Energy Sciences, under Contract No. DE-AC02-06CH11357. This research used resources of the Argonne Leadership Computing Facility, a U.S. Department of Energy (DOE) Office of Science user facility at Argonne National Laboratory and is based on research supported by the U.S. DOE Office of Science-Advanced Scientific Computing Research Program, under Contract No. DE-AC02-06CH11357. M.J.C. and H.C. also acknowledge support from the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences Data, Artificial Intelligence, and Machine Learning at DOE Scientific User Facilities program under Award Number 34532. GOVERNMENT LICENSE: The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory ("Argonne"). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan. http://energy.gov/downloads/doe-public-access-plan. The scanning probe microscopy research was supported by the Center for Nanophase Materials Sciences (CNMS), which is a US Department of Energy, Office of Science User Facility at Oak Ridge National Laboratory.

Author contributions

M.P., H.C., A.V., T.Z., V.S., and M.C. developed the context and tool-retrieval framework. M.P., H.C., A.V., T.Z., V.S., Y.L., R.V. and M.C. evaluated the model on domain specific questions. M.D. developed the interface to GPT3.5Turbo and R.H. interfaced the framework with the instrument. All authors discussed the results and contributed to the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41524-024-01423-2.

Correspondence and requests for materials should be addressed to Mathew J. Cherukara.