NealAI: A RAG-based Chat Assistant Supporting Command of a Robotic Team from the ISS

Philipp G. Knestel[1], Luisa Mayershofer[1], Anne Köpken[1], Nesrine Batti[1], Florian S. Lay[1], Timo Bachmann[1], Sant Brinkman[2], Jörg Butterfaß[1], Emiel Den Exter[2], Tristan Ehlert[1], Werner Friedl[1], Thomas Gumpert[1], Xiaozhou Luo[1], Ajithkumar N. Manaparampil[1], Tai Mai[1], Antoin Raffin[1], Florian Schmidt[1], Annika Schmidt[1], Daniel Seidel[1], Lioba Schürmann[1], Nicole Wenzinger[2], Bernhard Weber[1], Alin Albu-Schäffer[1,3], Adrian S. Bauer[1], Daniel Leidner[1,4], Rute Luz[2], Peter Schmaus[1], Freek Stulp[1], Thomas Krüger[2], Samuel Bustamante[*,1,3], Neal Y. Lii[*,1]

***Abstract*— We introduce NealAI, the first AI chat assistant to support astronauts with question answering during a space telerobotics experiment. In the Surface Avatar mission, an ISS crew member controlled a heterogeneous team of four robots in a simulated Martian environment. NealAI uses a Retrieval-Augmented Generation (RAG) approach, enabling a Large Language Model (LLM) to dynamically retrieve relevant context about the experiment and its robots, and deliver accurate, context-aware responses. To adhere to privacy requirements and computational costs, NealAI is based on a single small-scale LLM running locally. We assessed NealAI's performance in different evaluations, including a preliminary experiment with an ISS crew member teleoperating the robots, as well as a set of offline tests to evaluate the LLM context selection, the response correctness, and when (and why) hallucinations occur. Results demonstrate the feasibility and limitations of using a small-scale LLM on a RAG-based chat assistant during a space telerobotic experiment. Finally, we report some conclusions and lessons learned.**

I. Introduction

Progress in Lunar and Martian exploration has brought a need for robots to be deployed in-situ for resource utilization and local infrastructure support. Future missions will require robot teams with complementary capabilities, as well as user interfaces for humans to understand and command them effectively. Our DLR-ESA Surface Avatar Telerobtic Experiment [1], conducted between 2022 and 2025 aimed to develop and test new technology enabling a member of the International Space Station (ISS) crew to command a heterogeneous team of robots on ground, including a humanoid, two quadrupeds and a rover. These robots operate on different levels of autonomy, and perform tasks for environmental investigation and sample return.

Managing such a diverse robotic team requires a high cognitive load. Despite thorough pre-mission training, astronauts may forget specific instructions or lose the overview of the robots' actions, the mission goals or the user interface (UI) features, so they would benefit from an AI assistant to support them, for instance in natural language. Furthermore, as future missions target more remote environments such as

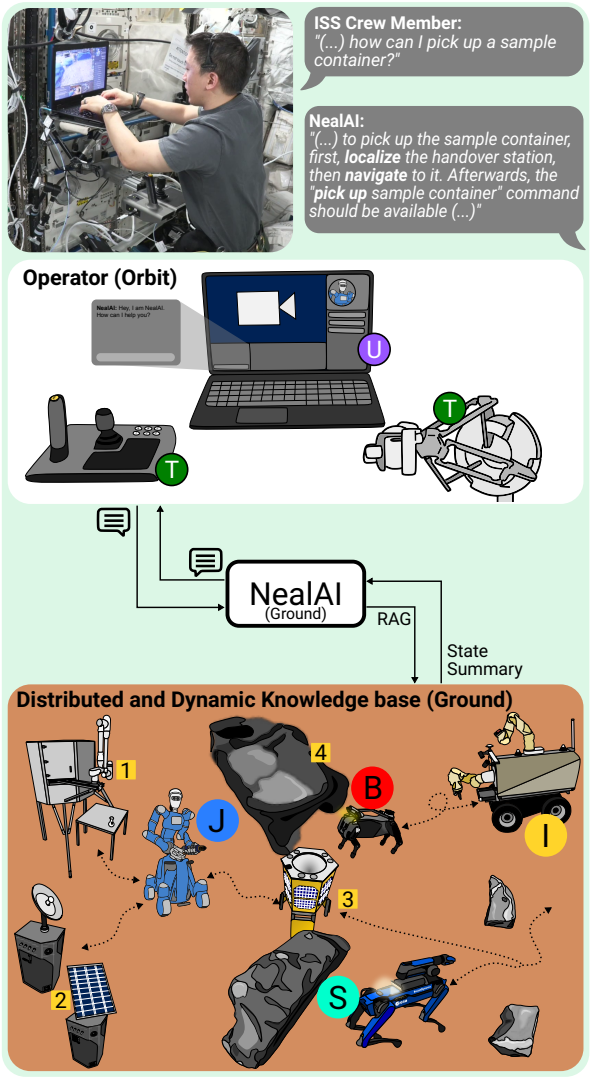

Fig. 1: NealAI overview deployed for the Surface Avatar mission. The crew member communicates with NealAI via a built-in chat panel; a snippet from a real interaction is shown. Mission components (see Section III) include four robots—the humanoid Justin (J), quadrupeds Bert (B) and Spot (S), and the Interact (I) rover—and four objects: (1) ELAFANT lander, (2) SPUs, (3) handover station, and (4) cave.

[1] German Aerospace Center (DLR), Robotics and Mechatronics Center (RMC), Münchener Str. 20, 82234 Weßling, Germany.
[2] European Space Agency (ESA), Human Robotics Interaction Lab (HRI), Keplerlaan 1, 2201 AZ Noordwijk, Netherlands
[3] Technical University of Munich, School of Computation, Information and Technology, Garching, Germany.
[4] University Bremen, Institute for Artificial Intelligence, Am Fallturm 1, 28359 Bremen, Germany.
[*] Equal contribution on a supervisory role.
We thank the German Space Operations Center, the Columbus Control Centre, and the European Astronaut Training Centre for the support during experiment preparation, testing, and astronaut training.

 DOI: No doi yet - to be presented in iSpaRo 2025

Mars, support from humans on ground will be limited. To address these challenges, in this letter we introduce **NealAI**[1], an AI workflow [2] prototype tailored for the third DLR & ESA Surface Avatar Session, illustrated in Fig. 1. NealAI uses an LLM to reply to astronaut questions about mission goals, robot commands and UI. As the internal robot beliefs are subject to change within the mission, NealAI obtains the mission context through RAG [3] method based on LLM tool usage. As the state of the robotic assets is constantly evolving during the missions, NealAI retrieves its context by generating summaries of the robot's state at query time [4]. Furthermore, due to strict privacy requirements, NealAI runs entirely on local LLMs beside the robotic systems. To prepare for limited resources in future on-orbit missions, it employs a small-scale model.

Our focus is not only on whether it is possible to design a RAG workflow with a relatively small language model. We also conduct an empirical study with two goals: (1) identifying the strengths and limitations of the method, particularly where it comes to so-called "hallucinations", i.e. factually wrong responses. And (2) identifying relevant question types during operation of the robots. With this goal, we conducted a preliminary experiment with an ISS crew member commanding the robots on sample return and planetary inspection tasks. We further conduced an offline evaluation of the system with a digital twin of the robots. We used the experimental results to draw some lessons learned for future RAG-based assistants in a space context, summarized at the end of the paper.

To summarize our contribution, (i) we present NealAI, the first on-orbit *RAG based question-answering LLM workflow for complex telerobotic space missions with a distributed and dynamic knowledge base about the robots state, robots control, user interface and mission protocol*. NealAI categorizes the astronauts' questions, extracts question-related information, and provides an answer; (ii) an offline experiment studies NealAI's capabilities and limitations; and (iii) an online experiment with an ISS crew member asking questions to NealAI while teleoperating the robots.

II. Related Work

Previous efforts to deploy AI assistants aboard the International Space Station (ISS) include CIMON (Crew Interactive Mobile Companion [5], [6], which utilized IBM Watson's rule-based AI [7] to assist astronauts during structured tasks. Since the advent of LLMs, there have been reports about their in space in the context of the ISS [8], but literature of their developments and findings is scarce. One such assistant systems is the Mars Exploration Telemetry-Driven Information System (METIS) which is being developed to conduct autonomous spacecraft operations and monitoring in human missions [9]. To increase reliability in knowledge-intensive tasks, it has been suggested [10] to enhance AI assistants like METIS with knowledge representations and retrieval-augmented generation (RAG, [3]) methods. Our paper builds upon RACCOON, our previous work for explainability in assistive robotics [4]. RACCOON is a framework for question-answering with two steps: first the required modules to answer a user question are selected given by an embeddings-based retrieval framework. Then, information by the robot is retrieved with so-called **state summaries** of the robot modules, which summarize the robot beliefs.

LLM tools: NealAI retrieves the state summaries by following a so-called *LLM workflow* paradigm, defined by Schluntz and Zhang as "*systems where LLMs and tools are orchestrated through predefined code paths*" [2]. Here, tools reference a standard LLM paradigm for allowing the model to call external functions made available to it [11], [12], which we use to enable the LLM to request information from any mission component. For example, if the astronaut asks about their goals, the LLM may call a tool with the day's protocol. Note that a tools-based system does not utilize embeddings-based reasoning, unlike traditional RAG or RACCOON. This choice improves scalability, as RACCOON would require a labeled dataset with example queries for every query type the system can handle, which becomes difficult to maintain.

III. Background: mission overview

NealAI was introduced in the third DLR-ESA Surface Avatar Prime Session, where an ISS crew member takes command over a heterogeneous robot team located at the German Space Operations Center in Wessling on July 21-24, 2025. Fig. 1 depicts the operator side, the NealAI interface, and the Mars environment with all relevant assets. In the "Distributed and Dynamic Knowledge base" part, the **Mars environment** and mission assets are shown. Four distinct robots participate in the mission:

- **Rollin' Justin (J)** [13], a humanoid robot.
- **Bert (B)** [14], a small-sized quadruped for exploration of constrained areas such as caves.
- **Spot (S)**[2], a large quadruped also equipped with a manipulator.
- **Interact (I)** [15], a rover platform with a manipulator.

These robots perform collaborative tasks such as collecting sample containers for return and exploring the environment. The astronaut is able to issue commands on different levels of autonomy, from supervised autonomy to direct teleopration [1]. The Martian environment includes:

- **ELAFANT (1)**: a robotized lander for inspecting and storing samples, and a camera platform.
- **SPU (2)**: Smart Payload Units simulating communication and energy modules
- **Handover Station (3)**: a location where robots exchange sample containers.
- **Cave (4)**: an exploration area for BERT
- **Watchtower** (not depicted): a camera platform.

On the **operator side** of Fig. 1, the astronaut aboard the ISS interactes with the mission assets through a laptop running a mission-specific knowledge-driven UI called OperatorUI [1] (marked as **U** in the figure). The GUI enables the

[1]The name is inspired by our principal investigator Neal Y. Lii, who supports our astronauts on the voice loop during the Surface Avatar missions.

[2]https://bostondynamics.com/products/spot/, last accessed 06.08.2025

astronaut to command and monitor the robots and mission environment, switching control between robots and managing operations concurrently. The GUI also enables chats with NealAI, using a chat panel and keyboard input. In order to teleoperate the robots, the astronaut can use a joystick and a 7 DoF force reflection input device (SIGMA)(sigma.7, Force Dimension, Nyon, Switzerland) [3] to control the robot cameras, manipulators and mobile bases. The Joystick and the SIGMA are marked with **T** (teleoperation) in figure 1.

IV. NealAI: an LLM-based question-answering workflow.

NealAI must handle information from multiple sources, such as the OperatorUI, the mission protocol, and the robotic systems, each with distinct characteristics and requirements. One aspect of this challenge is the separation between components: questions about the OperatorUI and mission protocol can be answered with standard documentation, while the robots operate on different systems, possess distinct world models [16] and belief systems, and involve unique tasks, commands, and error-handling processes [17]. Thus, the information is of distributed nature, and can be dynamic as the robot states evolve continuously during a session as robots interact with the environment and receive user inputs. To give an example, our humanoid robot Rollin' Justin may or may not enable the astronaut to grasp a sample container at a given time, for instance if the object requires localization first. Our RAG system must capture this dynamic information, in order to provide not only contextually relevant responses, but also *up-to-date* responses.

We propose a multi-step tools-based LLM workflow based on a local LLM. We enable the LLM to take decisions and request information by using so-called structured tools (e.g. [11], [12]), i.e., the ability to serialize functions within an Application Programming Interface (API), and call them via text, a task for which modern LLMs are trained. In our case, the API is a set of functions that we provide for retrieving information from the different assets in the system, obtaining a "retrieved context", i.e., text snippets that contain relevant information about the query [4], such as summaries from the robot internal models. Specifically, our query-answering framework consists of three steps: selecting information sources to retrieve information; obtaining a text snippet from them; and using them to generate an answer for the user in natural language, also using an LLM. The steps are described in detail below.

A. Step I: selecting the information source

As illustrated in Fig. 2, the LLM's goal is to select one of five mission-related classes based on the prompt: Specific Robot Action (questions about actions in the dynamic state summary), General (about the robot, mission, or situation), Teleoperation (how to teleoperate the current robot), OperatorUI (the astronaut GUI), and Mission Protocol (the mission's procedural).

To aid the LLM, we provide a prompt with additional context, also shown in Fig. 2. To construct the prompt, we

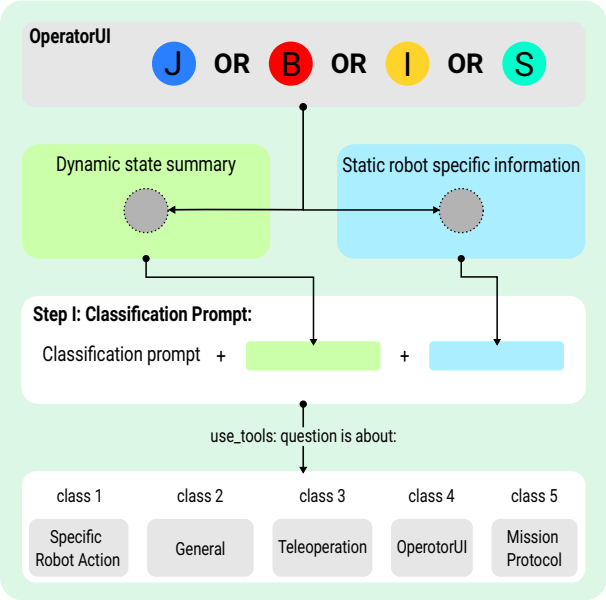

Fig. 2: Visualization of the Step I prompt showing the selected robot in the OperatorUI—Justin (J), Bert (B), Interact (I), or Spot (S)—and the context using the Dynamic State Summary (green) and Static Robot Information (blue). Also shown are the class categories used in the RAG-based tool-selection process.

first obtain the robot being operated by the astronaut in the UI, which provides the robot's name and specific contextual information. We then generate three text snippets to assist the LLM in tool selection: (1) a static description of the selected robot (blue in Fig. 2); (2) a dynamic state summary[4] listing the actions the robot can currently execute (green in Fig. 2); and (3) the general mission context and the role of NealAI. We study the effect of this additional context in the experiments in Section V-D. Finally, the prompt also explains the tool selection task.

B. Step II: retrieving a specialized state summary

After the LLM categorizes the question in Step I, we aim to create a text snippet that includes ground truth information specific to the astronaut query, and which will be used in the following step to generate an answer. This is analogous to summarizing context chunks in classical RAG frameworks, and is illustrated in the pink snippets in Fig. 3.

This **specific context for the question** depends on the categorization from Step I. For each categorization, different information is provided to the LLM as shown in the figure. We distinguish two types of prompts: static text i.e., information that does not change during the mission; and dynamic text, i.e information that changes during the mission, and has to be retrieved accordingly.

1) Retrieving dynamic snippets for specific robot actions: In our mission, we use dynamic snippets to provide the astronaut with instructions about **a specific robot action**. For instance, if the astronaut wants instructions for executing an action, we find out if the action can only be executed after certain prerequisites, and if so, derive instructions for it.

The procedure to retrieve this dynamic snippet is summarized in Fig. 4, including an example. First, we collect the set

[3]https://www.forcedimension.com/products/sigma, last accessed 06.08.2025

[4]Not to be confused with the *retrieved* state summary in the next step

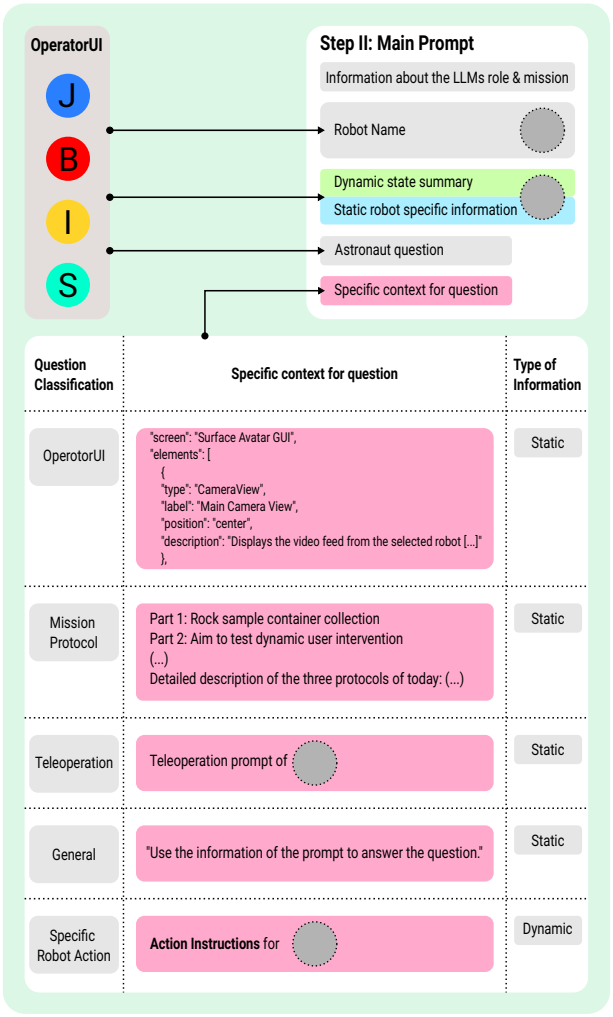

Fig. 3: The Step II prompt includes the selected robot in the OperatorUI; the context using the Dynamic State Summary (green) and Static Robot Information (blue); the astronaut's question. The table shows the specific ground-truth context for each class category (pink).

of all actions that the astronaut can execute for the current robot during the mission. These actions could be either *available*, meaning that it's preconditions (e.g. localization) are met, and the action can be issued right away; or *blocked*, meaning the action would require instructions. This list varies for each robot due to their unique capabilities.

The process begins with the LLM receiving the astronaut's question and the list of all available and blocked action names. The LLM matches the question to one action from the list, using RAG selection. If this action is available, the returned context is simply "*[action] is directly available in the OperatorUI*". However, if the action is blocked, the next steps depend on the robot. For example, with Spot, actions such as *dock* and *undock* cancel each other out. If Spot is currently docked or not localized, actions like *move to handover* are blocked. Interacts provided actions are implemented in a similar way. There are only primary reasons that cause an action for Interact or Spot to be blocked. Consequently, a dictionary-based method is employed to identify and return the specific reason why the requested action is blocked.

In contrast, the commanding approach of Justin is more complex, the robot uses a symbolic planner to achieve a

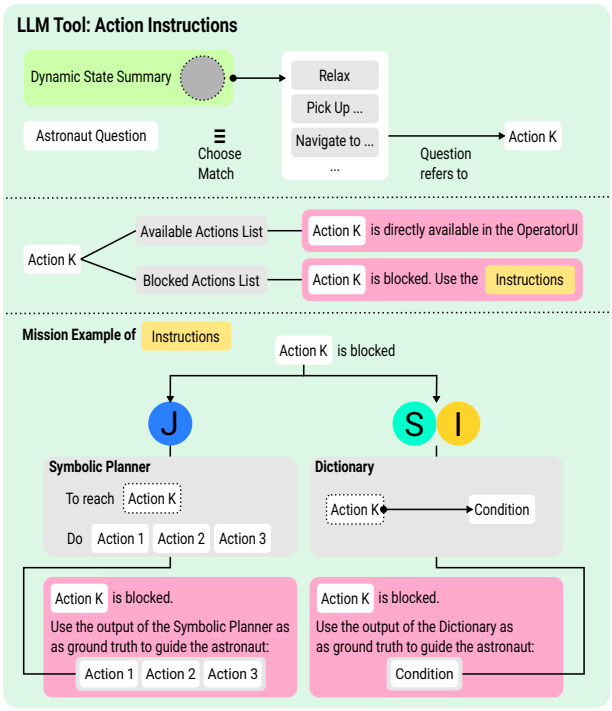

Fig. 4: From the Dynamic State Summary (green), the LLM selects the action matching the question. Blocked or available actions yield different ground-truth context (pink). In the Surface Avatar mission, this covers Justin (J), Spot (S), and Interact (I): a symbolic planner generates valid paths for Justin, while a dictionary provides context for Spot and Interact.

desired goal state. [18]. For instance, to pick up an object on a table, the robot needs to have located the object and have navigated to the table, and for this the table must be located first. To provide instructions, we use the symbolic planner to compute and return a *plan* that solves the preconditions of the specific action requested by the astronaut.

2) Retrieving static snippets: Static types of categories include *Mission protocol*, *OperatorUI*, *Teleoperation*, and *General*. The snippet includes static information that does not change over the course of the mission. Specifically:

- **OperatorUI**: a detailed description of the Operator's GUI, including positions on the screen. Example query: ``How do I switch robots in the UI?''.
- **Teleoperation**: robot-specific explanation for operating the cameras and manipulators. Example: ``How do I move the robot camera?''.
- **General**: No specific context is added to the summary in Fig. 3. Ex: ``List the robot commands.''.
- **Mission protocol**: A summary of the astronaut goals. Example: ``What are today's goals? ''.

C. Step III: providing an answer

Last, the LLM is queried to provide an answer in natural language (i.e., without using the tools) given the retrieved context. The prompt shown in Fig. 3 consists of several parts: Firstly, a short text describes the LLM's function as an assistant for the astronaut and the mission design, which involves multiple robots controlled by the astronaut from the ISS using the OperatorUI. To provide a "safety net" in case of a failed retrieval, we also provide the "general" dynamic state summary from Step I (the green snippet in Fig. 2). Finally,

the astronaut's question and the specific context retrieved in Step II are added to the prompt. An example final generation from our ISS session is shown in Fig. 1. Here, the LLM used the "Mission protocol" tool.

a) Guardrails: We implement additional checks to verify that the LLM's response stays within the scope of the mission. Taking inspiration from the LLM community, we do this with a further "LLM hop" [19] with a description of the mission and relevant topics. The original answer generated by the LLM is also included in this prompt, but not the astronaut's query, such that the LLM self-checks its own response. While this guardrails the system for the mission, it entails that the LLM is blocked from answering general-purpose question, making it less interactive and restraining it from being useful outside of mission contexts. This can limit the interactions. During our ISS session, for instance, the astronaut tried to test the system with arithmetic questions, but the system filtered out these answers.

D. Implementational notes

We utilized the Mistral Small 3.2 model [20], a lightweight model, served via Ollama on a system with dual 48GB NVIDIA RTX 6000 Ada GPUs. We employed Ollama's default quantized model, which ran significantly faster than the full-precision version without noticeable performance loss, with a mean inference time of 4.5 seconds. As preliminary tests showed reduced answer quality with longer contexts, we disabled follow-up queries, meaning crew queries could not access previous chatbot interactions.

V. Offline Evaluation with Digital Twins

In order to get a more in-depth look at NealAI's performance, we conducted an offline assessment with digital twins of the robots on the same artificial martian environment as the ISS crew member. We conducted four experiments, presented in Sections V-B to V-E$_s$: the first three parts evaluate the automated tool-selection accuracy, and the fourth part the end-to-end correctness of the answers.

A. Data description

1) Query datasets: In order to evaluate the system, we collected datasets with queries about the mission. A first evaluation dataset $\mathcal{D}_1$ consists of a set of queries provided by the Surface Avatar team members (all of them in the authors list), and extended with a set of questions the ISS crew member asked during the online experiment, described in Section VI. $\mathcal{D}_1$ contains a total of 161 questions, including questions about the OperatorUI (14), teleoperation (22), mission protocol (20), general topics (23), and specific robot actions (82). The questions about specific robot actions are distributed between robots as follows: Justin (31), Bert (9), Spot (26), and Interact (16). To obtain more data, we also augmented the queries by asking an LLM to rephrase them and return different formulations without changing the meaning nor adding new information. To reduce LLM bias, we used a different LLM for data generation (ChatGPT, OpenAI, San Francisco, California). To evaluate different aspects of NealAI, the dataset was split into $\mathcal{D}_{\text{augmented\_1}}$, containing 110 questions on OperatorUI, teleoperation, mission protocol, and general topics, but not specific robot actions; and $\mathcal{D}_{\text{augmented\_2}}$, containing 309 questions about robot actions (some of them involving the objects in the environment) distributed along the robots. Each question is labeled with a ground truth tool-selction classes provided by a human expert (in the authors list).

2) NealAI answer scenarios : In order to provide NealAI answers to the questions in $\mathcal{D}_{\text{augmented\_1}}$, $\mathcal{D}_{\text{augmented\_2}}$ and $\mathcal{D}_1$ we evaluated the system across all four robots. Because the environment and robots evolve over time through astronaut manipulation and control, each evaluation part is conducted across four distinct scenarios derived from the surface avatar artificial martian environments.

Each scenario below is evaluated from different robot perspectives, considering all five question classifications established in the Phase I prompt (see Fig. 2).

- *Scenario 1*: Justin is not localized, Bert is deactivated, Interact operates normally, and Spot is docked to charge.
- *Scenario 2*: Justin localized to the handover station, Bert is deactivated, Interact positions Bert in front of the cave, and Spot autonomously searches for sample containers.
- *Scenario 3*: Justin localized to ELAFANT, Bert operates normally while inspecting the environment, and both Interact operates normally and Spot lost localization.
- *Scenario 4*: Justin localized to SPU3, Bert unable to walk due to malfunction, Interact and Spot operate normally.

B. Exp. 1: Tool-selection accuracy of non-action questions

Can NealAI correctly classify queries about the operator GUI, teleoperation, the mission protocol and surface avatar in general? We asked NealAI each question of $\mathcal{D}_{\text{augmented\_1}}$ twice for each robot's selection, and for all four scenarios, a total of $110 * 4 * 4 * 2 = 3520$ interactions. From these, 89 (2.5%) were incorrectly flagged as off-topic, and discarded. Thus this resulted in 3,431 pairs containing a ground-truth class and NealAI's classification result. Fig. 5 summarizes the accuracy of NealAI's categorization.

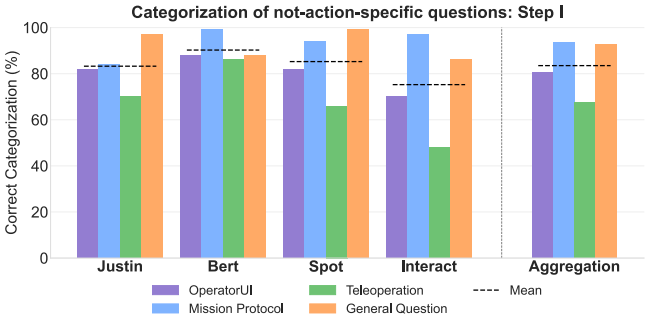

Fig. 5: Tool-selection accuracy for $\mathcal{D}_{\text{augmented\_1}}$ across all robots and scenarios (totaling 3,431 questions). Accuracy is presented for each robot individually as well as aggregated.

Discussion: The aggregated results indicate an overall accuracy of 84%, with performance on OperatorUI and Teleoperation questions falling below this average. When NealAI selects an incorrect tool, it predominantly defaults

to the general tool, leading to a lack of knowledge in Step II to address the question. Additionally, questions about the robots Justin and Interact exhibit lower performance compared to Bert and Spot. Given that the question sets are identical across robots, these discrepancies can be attributed to differences in the dynamic state summaries and robot-specific information incorporated into the Phase I prompt (see Fig. 2). This suggests that the additional information may introduce noise, affecting the tool-selection process - further evaluated in Section V-D.

C. Exp. 2: Tool-selection accuracy of action specific questions

Can NealAI, using the Step I prompt in Fig. 2, correctly classify queries about robot-specific actions? Can NealAI choose the action according to the query? We evaluated (1) the tool-selection accuracy for the class *specific robot action*, and (2) the accuracy of selecting the correct action within the robot's overall list of actions (see Fig. 4). We asked each question twice for all four scenarios, noting that the questions were robot-specific, thus yielding a total of $309 * 4 * 2 = 2472$ interactions. From these, 38 (1.5%) were incorrectly flagged as off-topic, and discarded, yielding 2,434 tuples including the ground-truth class (robot-specific actions), the ground-truth action the query is referring to, NealAI's Step-I classification, and Neal AI's action selected within the pool of possible robot actions. We report NealAI's accuracy in Fig. 6.

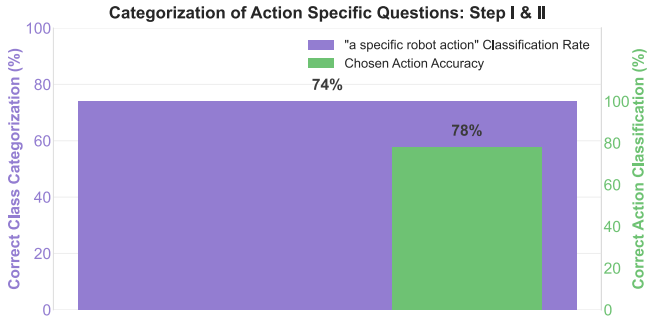

Fig. 6: Tool-selection accuracy for the *specific robot action* class, alongside the accuracy of selecting the correct action. Correct action selection is conditional on accurate initial classification.

Discussion: The overall tool-selection accuracy of Step I for the class *specific robot action* is 74%, which is lower compared to other classes, given that (as in the previous experiments) NealAI defaults to the general tool. If NealAI selects "robot specific actions", the accuracy in the action matching step is 78%, resulting in an overall accuracy of 57.7%. For instance, a question regarding how to pick the sample container should be classified as *specific robot action* with the corresponding action *pick up sample container*. This does not imply that NealAI fails to provide an answer, as a dynamic state summary is repeated in Step II (see prompt in Fig. 3), and NealAI could still reply the action is available or blocked. However, for blocked actions, accurate tool selection is critical to accessing the key information needed to resolve the astronaut's issue.

D. Exp. 3: Step I prompt ablation

What is the effect of the additional robot-specific context in Step I's prompt? As shown in Figure 2, we add to the Step I prompt snippets containing a *Dynamic State Summary* (green) and *Static Robot-Specific Information* (blue). To study the effect of this prompt, we repeat the experiments in Sections V-B and V-C under the same conditions, with the only difference being the prompt ablated these two snippets. The results in Table I summarize the results aggregated by robot, where the first row summarizes the results from Figs. 5 and 6, and the second row presents the new results with the ablated prompt.

	General	Teleoperation	Mission Protocol	OperatorUI	Specific Robot Actions
With context	82%	71%	89%	74%	**74%**
Without context	**93%**	**90%**	**97%**	**92%**	48%

TABLE I: Evaluation results of Step I tool-selection accuracy with and without context (*Dynamic State Summary* and *Static Robot Specific Information*) in %

Discussion: as Table I shows, all tools except **Specific Robot Actions** perform better when the context is omitted from the Step I prompt during tool selection. Compared to the strategy including context, this approach achieves a 6% absolute increase in overall tool-selection accuracy. The ablated prompt snippets (*Dynamic State Summary* and *Static Robot-Specific Information* act as a bias towards making the LLM specifically answer questions about robot specific actions. Thus, using can lead NealAI to be more accurate on a specific mission area selected.

E. Exp. 4: Human expert judging generation quality

Can NealAI generate truthful responses? Can it recover and produce correct answers, even with the wrong context? $\mathcal{D}_1$ questions on OperatorUI, teleoperation, mission protocol, and general topics are asked from the perspectives of each of the four robots, while questions related to *robot-specific actions* are asked on one specific robot, as described in Section V-A.1, resulting in a total of 387 questions. A human expert (in the authors list) was provided with the question and NealAI's response, without access to information about the tools used. An answer was considered *correct* if it was free of hallucinations and contained ground truth information relevant to the question. We report the results in Fig. 7, and we further split the answers whether the correct tool was chosen or not.

Discussion: Aggregated results in Fig. 7 indicate that NealAI answers a large percentage of the queries with truthful answers without hallucinations (76%). As expected in a RAG system, correct answers are highly correlated on correct context (tool) selection, as this already predicts 86% of the correct answers.

But why did NealAI still answer 43% of wrongly-sorted questions correctly? An analysis by category reveals distinct patterns. As illustrated in Figure fig. 3, the Step III prompt includes the dynamic state summary and robot-specific information, providing NealAI with knowledge of available and

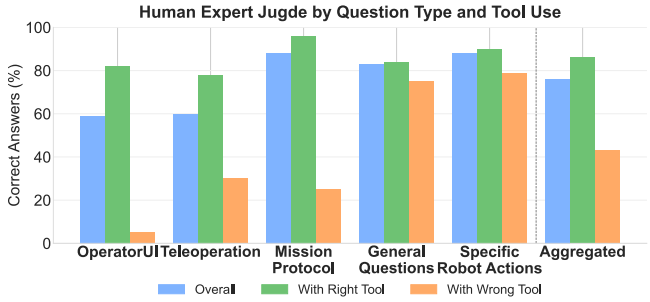

Fig. 7: Human expert evaluation of $\mathcal{D}_1$ where answers are classified as *correct* or *not correct*. The figure displays the percentage of overall correct answers per category, along with the accuracy of correct answers given the use of the right or wrong tool.

blocked actions, as well as general details about the robot and mission. For questions about available actions, the information remains largely constant across scenarios (see Fig. 3). A similar situation applies to general questions, which do get additional contextual data provided. Consequently, NealAI can often produce correct answers to general questions or questions regarding specific robot actions despite incorrect tool selection. This explains the relatively small difference in accuracy between correct answers given with the right or wrong tool selection for both the *specific robot action* and *general* categories. In contrast, questions concerning OperatorUI, Teleoperation, and Mission Protocol demonstrate a strong dependence on accurate tool selection. Notably, the correctness rate for OperatorUI questions answered with the wrong tool is only 5%. This low rate reflects that the source of UI information available to NealAI is accessed via the correct tool. We refer to this in the discussion in Section VII.

VI. Online evaluation with an astronaut

The evaluation presented is complemented by qualitative feedback gathered from an ISS crew member who asked queries to NealAI from space, while teleoperating the robots on Surface Avatar's Prime Session 3. The real-world deployment provides insights into the system's practical strengths and limitations that extend beyond quantitative metrics. During the session, the astronaut[5] was given five predefined questions ask "in his own words". These included one query about the mission protocol, a general question on the robotic team's capabilities, one regarding blocked actions of the robot Justin, instructions for unblocking these actions, and an open question on NealAI's general assistance capabilities. These queries aligned well with the available tools and, as expected, received the required ground-truth information to produce accurate responses.

Following this, the astronaut was given some minutes to ask unstructured, open world questions with the system, revealing several key strengths and limitations of NealAI:

1) For general knowledge questions like *"Which robots resemble dogs?"*, NealAI correctly identified Bert and Spot as quadruped robots. However, the subsequent question *"Which robot can move the fastest?"* was incorrectly answered with *"Bert, the quadruped robot designed for exploration, can move the fastest."*
2) The follow-up question *"And how fast is that?"* was filtered as out of scope due to the lack of chat history integration.
3) A question about UI issues, *"How to realign the overlays?"*, was misclassified in Step I, resulting in an hallucinated answer.
4) Further questions about unavailable information, such as the speed of Bert or which robot has the most cameras, were correctly answered with responses indicating that the information was not available to NealAI.

After interacting with NealAI, the ISS crew member completed a questionnaire to rate the system's helpfulness and knowledgeability, providing the answers in Table II. The crew member was asked to rate the knowledge and the helfpulness of the system on a scale from 0 (less knowledgeable/less helpful) to 6 (more knowledgeable, more helpful), where we note that 3 corresponds to the mid-scale.

		Was the AI assistant helpful in providing appropriate information about:			
	Knowledge*	**General**	**OperatorUI**	**Robots**	**Mission**
Astronaut	3.00	5.00	4.00	4.00	3.00

TABLE II: Post-session questionnaire ratings (0–6 scale). *) Question: Was the AI assistant sufficiently knowledgeable about the mission?

The astronaut's ratings ranged from 3–5, with highest for *General* (5) and lower for *Knowledge* and *Mission* (3).

VII. Discussion, findings, lessons learned

This paper's evaluation combines both offline experiments and a real-world online deployment with an ISS crew member, providing complementary insights into NealAI's capabilities and limitations. Overal, the results show that NealAI is able to retrieve context from a complex and distributed knowledge base (from 71% to 97% in Table I, depending on the conditions), and generate end-to-end answers to queries (76% in Fig. 7), even arbitrary, open-world ones from an ISS crew member with above-average qualitative astronaut rating (Table II). Nevertheless, we note two main limitations:

First, the tool selection accuracy (e.g., for the *Teleoperation* class in Experiment 1, see Section V-B) shows high variance, which can lead to hallucinations. This partly stems from the *Dynamic State Summary*, whose action names may confuse the LLM, causing misclassifications in Step I (e.g., interpreting teleoperation as an action). As shown in the ablation of Section V-D, teleoperation accuracy improves from 71% to 90% when this summary is removed. This underscores a limitation in LLM tool selection accuracy. We hypothesize that handling such specific contexts may be constrained by the small model size, and plan to evaluate a larger LLM in future work. For scale, the largest open-weights LLM, Kimi-K2, has 1 trillion parameters—about 41.7× larger than Mistral Small 3.2.

Second, the tool-selection accuracy in Step I differs significantly between Experiments 1 and 2 compared to Experiment 3 (see Table I). Notably, the accuracy for *Specific Robot*

[5]The ISS crew seemed to be eager to interact with the AI system, and expressed he wanted to explore the system.

Actions questions drops to 48%, representing a trade-off between including or excluding larger context in the Step I prompt. Omitting the action list reduces the likelihood that questions align with the *Specific Robot Action* classification space. However, as shown in Figure V-E, the overall accuracy for *Specific Robot Actions* answers remains at 79% even if the tool is wrongly selected. Therefore, a lesson learned is that omitting the dynamic state summary results in a better performance of NealAI in general (6% increase in the overall tool-selection accuracy.), with only loses for specific robot actions. However, this would entail that detailed *Action Instructions* would not be provided in some cases.

The online session with the ISS crew member confirmed many of these findings. Key observations from both evaluations include:

- NealAI effectively filters out-of-scope questions, but this filtering can restrict user interaction beyond mission-related topics, as noted by the astronaut.
- Misclassification of certain question types leads to hallucinated or incorrect responses, as seen in the overlay realignment query.
- The lack of chat history integration limits the system's ability to contextualize follow-up questions, affecting conversational flow and user satisfaction. However, adding history would add more context, which could reduce the system's overal accuracy.
- The online evaluation showed that NealAI can correctly indicate when information is unavailable, though it occasionally produces hallucinated answers.

These limitations contributed to the astronaut's assessment of NealAI's overall mission knowledge as moderate (about 3 on a 0–6 scale). In contrast, NealAI scored higher for providing general information, robot-specific data, and OperatorUI-related responses, according to the astronaut's questionnaire feedback. The findings highlight the need for future work on improved question classification, where traditional search-based knowledge and embedding-based retrieval could enhance context selection. Finally, the qualitative results should be validated with a larger user population, for example through a multi-subject user study.

VIII. Conclusion

NealAI supports commanding a heterogeneous robotic team by identifying key question types and providing ground truth information to the astronaut, as demonstrated in offline and ISS crew experiments. However, the limitations of this early prototype include misclassification of open questions and hallucinations arising from queries about non-selected robots. Feedback from the ISS crew member and the user study offered valuable insights into user needs. Our experiments highlighted both strengths and limitations of the system. Future work should focus on improving question classification and enabling queries about all robots regardless of selection. Nonetheless, NealAI demonstrates that a small-scale LLM can effectively manage a RAG-based tool-selection approach for heterogeneous information, spanning robot internal states to the OperatorUI. This was further supported by the astronaut's hands-on interaction with the system.

References

[1] N. Y. Lii, P. Schmaus *et al.*, "Introduction to surface avatar: the first heterogeneous robotic team to be commanded with scalable autonomy from the iss," in *Proceedings of the International Astronautical Congress, IAC*, vol. IAC-22. International Astronautical Federation, IAF, September 2022. [Online]. Available: https://elib.dlr.de/189618/

[2] E. Schluntz and B. Zhang. (2024) Building effective agents. https://www.anthropic.com/engineering/building-effective-agents. Engineering at Anthropic blog. [Online]. Available: https://www.anthropic.com/engineering/building-effective-agents

[3] P. Lewis, E. Perez *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[4] S. Bustamante Gomez, M. W. Knauer *et al.*, "Raccoon: Grounding embodied question-answering with state summaries from existing robot modules," in *2025 IEEE International Conference on Robotics and Automation, ICRA 2025*. IEEE, 2025. [Online]. Available: https://elib.dlr.de/214144/

[5] Airbus, "Cimon-2 makes its successful debut on the iss," https://www.airbus.com/en/newsroom/press-releases/2020-04-cimon-2-makes-its-successful-debut-on-the-iss, Apr. 2020, accessed: 2025-07-30. [Online]. Available: https://www.airbus.com/en/newsroom/press-releases/2020-04-cimon-2-makes-its-successful-debut-on-the-iss

[6] T. Eisenberg, G. Reichert *et al.*, *CIMON – The First Artificial Crew Assistant in Space*, 2025, pp. 149–163.

[7] Y. Chen, J. D. Elenee Argentinis, and G. Weber, "Ibm watson: How cognitive computing can be applied to big data challenges in life sciences research," *Clinical Therapeutics*, vol. 38, no. 4, pp. 688–701, Apr. 2016, copyright © 2016 The Authors. Published by Elsevier Inc. All rights reserved. [Online]. Available: https://doi.org/10.1016/j.clinthera.2015.12.001

[8] B. A. Hamilton, "Deploying a large language model in space," https://www.boozallen.com/insights/ai-research/deploying-a-large-language-model-in-space.html, 2025, accessed: 2025-07-30.

[9] C. Hartmann, F. Speth *et al.*, "Metis: An ai assistant enabling autonomous spacecraft operations for human exploration missions," in *2024 IEEE Aerospace Conference, AERO 2024*. Institute of Electrical and Electronics Engineers (IEEE), May 2024. [Online]. Available: https://elib.dlr.de/210422/

[10] O. Bensch, L. Bensch *et al.*, "Towards a reliable offline personal ai assistant for long duration spaceflight," 2024. [Online]. Available: https://arxiv.org/abs/2410.16397

[11] T. Schick, J. Dwivedi-Yu *et al.*, "Toolformer: Language models can teach themselves to use tools," 2023. [Online]. Available: https://arxiv.org/abs/2302.04761

[12] Y. Qin, S. Liang *et al.*, "Toolllm: Facilitating large language models to master 16000+ real-world apis," 2023. [Online]. Available: https://arxiv.org/abs/2307.16789

[13] C. Borst, T. Wimbock *et al.*, "Rollin' justin - mobile platform with variable base," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 1597–1598.

[14] D. Seidel, A. Schmidt *et al.*, "Toward space exploration on legs: ISS-to-earth teleoperation experiments with a quadruped robot," in *Proceedings on IEEE Conference of Telepresence*. IEEE, 2024.

[15] T. Krueger, E. Ferreira *et al.*, "Designing and testing a robotic avatar for space-to-ground teleoperation: the developers' insights," in *71st International Astronautical Congress, IAC 2020*. International Astronautical Federation, 2020.

[16] R. Sakagami, F. S. Lay *et al.*, "Robotic world models – conceptualization, review, and engineering best practices," *Frontiers in Robotics and AI*, vol. 10, November 2023. [Online]. Available: https://elib.dlr.de/198741/

[17] N. Batti, L. Mayershofer *et al.*, "Toward intuitive robot-to-human error reporting to enhance user awareness in space (tele) operation," in *2025 IEEE Aerospace Conference*. IEEE, 2025, pp. 1–13.

[18] D. Leidner, A. Dietrich *et al.*, "Object-centered hybrid reasoning for whole-body mobile manipulation," in *2014 IEEE ICRA*. Hong Kong, China: IEEE, May 2014, pp. 1828–1835.

[19] M. Mathys. (2025, Jun.) You shall not pass: the spells behind gandalf. Lakera AI. Last updated: June 3, 2025. [Online]. Available: https://www.lakera.ai/blog/who-is-gandalf

[20] MistralAI, "Mistral-small-3.2-24b-instruct-2506," https://huggingface.co/mistralai/Mistral-Small-3.2-24B-Instruct-2506, 2023, accessed: 2025-07-30.