# Grounding Embodied Question-Answering with State Summaries from Existing Robot Modules

Samuel Bustamante, Markus Knauer, Jeremias Thun, Stefan Schneyer, Bernhard Weber, Freek Stulp

***Abstract*—Explainability in robotics is vital for establishing user trust. Recently, foundation models (e.g. vision-language models, VLMs) fostered a wave of embodied agents that answer arbitrary queries about their environment and their interactions with it. However, as VLMs answer queries based on camera images instead of on internal robot components, they cannot be applied directly to existing robot architectures which represent the robot's tasks, skills, and beliefs about the state of the world.**

**To overcome this limitation we propose RACCOON, a framework that combines foundation models' responses with a robot's internal knowledge. Inspired by Retrieval-Augmented Generation (RAG), RACCOON selects relevant context, retrieves information from the robot's state, and utilizes it to refine prompts for an LLM to answer questions accurately, bridging the gap between the model's adaptability and the robot's domain expertise.**

## I. Introduction

For users of assistive robots, explainability and transparency are central to fostering trust [8], [29]. One way to build explainable robots is by enabling them to articulate answers to user queries, based on their internal beliefs, in real time [8], and in natural language. In recent years this was made possible by deploying Foundation Models, such as Large- and Vision-Language Models (FMs/LLMs/VLMs), as language and vision interfaces for robots. Their abilities to answer open-vocabulary queries brought a number of embodied agents that describe their environment and their own actions based on camera images [22], [7], [25], [15], [10].

The aim of this work is to combine such approaches with existing robotics architectures, which commonly have software modules[1] which represent the robot's tasks, skills, and beliefs about the current state of the world [23], [17], [1]. Our motivation is that whenever FMs answer questions on behalf of the robots (e.g. based on camera images), they do not have access to the beliefs inside these modules. Thus they can only provide a post hoc interpretation of what the robot *could perceive in an image* or what it *seems to be doing*, instead of what it perceives and does. Consequently, these answers are not trustworthy with respect to the beliefs of the robots with pre-existing architectures for perception, planning and acting. In Fig. 1, for example, a VLM speaking for our assistive robot EDAN [28], might claim it locates a microwave if provided with a camera image containing one, even when EDAN's object locator missed it. The goal is therefore to create a system that grounds its answers on the robot modules, and not directly on the understanding it has from the world inside its weights.

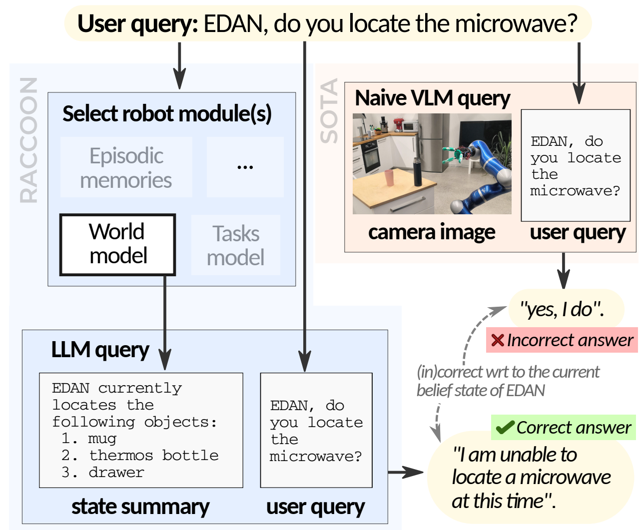


Fig. 1. An overview of the RACCOON framework (left), and a comparison with a VLM query as in the state of the art (SOTA) (right). RACCOON enables the internal state of existing robot modules to be taken into account when answering queries, leading to more truthful answers wrt to this state.

In this paper we propose RACCOON[3], a framework that grounds LLM responses in the robot's internal knowledge. Inspired by the Retrieval-Augmented Generation (RAG) community [11], our system retrieves so-called *state summaries* from existing robot modules, and uses them in a prompt for an LLM to answer. We thus take advantage of the flexibility offered by FMs while grounding their answers in the robot modules. Going back to the example in Fig. 1, RACCOON enables the robot to accurately explain that it does not locate the microwave. This example motivates for need for a tighter coupling between the FM and existing robotic modules.

To summarize our contribution, RACCOON is *a question-answering framework for robots with modules that represent the world, tasks and skills*, that: ($i$) selects the correct robot module(s) to be queried in order to answer a user question, inspired by information retrieval in RAG; ($ii$) extracts robot- and situation-specific state summaries from these modules; and ($iii$) provides this information to an LLM (with prompt engineering) to answer the question on behalf of the robot. The paper includes demonstrations and experiments on our robot EDAN.

All authors are with the German Aerospace Center (DLR), Robotics and Mechatronics Center (RMC), Münchner Str. 20, 82234 Weßling, Germany.

[1]Note that these modules can also be specialist FMs, e.g. models creating symbolic representations of task affordances [2] or the world state [12], [6].

[3]The acronym stands for "*Robots Answering questions grounded on COntextual cOgNitive modules*"

## II. Related Work

**Embodied question-answering for explainability:** Definitions of explainability for embodied agents in the literature refer to system's modules that explain their internal workings (such as their intents, policies and plans) to humans as their target [29]. A trend in the AI planning community is enabling robots to explain their plan decisions by answering so-called *contrastive questions* [21], i.e. queries of the form "`why did you decide on P / why did P happen and not Q?`" (e.g. [5], [3], [16], [26], [9]). Other post-hoc explanation techniques focus on accumulating and modeling knowledge during decision processes [24]. Our approach is not aimed to replace these methodologies. Rather, RACCOON tackles two problems: (1) that a prerequisite to construct explanations with existing methods is to parse input queries to match known templates and controlled vocabulary [27]; and (2) that there are many different modules within a complex robotic system (e.g. EDAN [28]), and thus there are arbitrarily many different types of questions that the system can answer from them. In this regard, our approach leverages RAG-LLMs to provide flexibility and rich natural language to the question-answering process.

**Retrieval Augmented Generation:** Tasks that require specific knowledge often pose challenges for LLMs, which may provide outdated responses or, if the topic is unknown to them, hallucinate responses. Retrieval-augmented generation (RAG) is a well-established technique that enhances the performance of LLMs in such knowledge-intensive tasks by augmenting their answers with query-related knowledge, retrieved from an external database [18], [14].

In *naive RAG* [11] for instance, an external text database is partitioned in so-called "document chunks" $x_{i=1:N}$. Off-line, a text embedding model $\mathcal{E}$ is used to convert each chunk into a vector $\boldsymbol{y}_i \in \mathbb{R}^m$, where $m$ is a parameter of the text embedding model. The result is a *vector store* $\{(x_i, \boldsymbol{y}_i)\}_{i=1}^N$. On-line during inference, a user query $x_u$ is converted to the same vector space $\boldsymbol{y}_u = \mathcal{E}(x_u)$, and $\boldsymbol{y}_u$ is used to extract relevant document chunks, based on their proximity in the vector space. Crucial for this approach is the fact that sentences with similar meaning tend to be close in vector space. The selected document chunks are then used as context information in the prompt. This allows RAG to take into account information from the external database, i.e. information which may not be represented in the LLM itself.

## III. RACCOON

The goal of RACCOON is to obtain contextual information about the current state of existing robot modules for an LLM to answer a user query. An example is the prompt in Listing 1. The user query "`EDAN, can you locate the microwave?`" is accompanied by the context that EDAN currently only locates a mug and a thermos bottle. The context snippet is a state summary provided by the worldmodel module. This allows the LLM to take the worldmodel state into account when answering the user query.

```
;; Preamble: explain the LLM the goals
You are the speech AI module of the robot (...)
Your goal is to answer questions (...)
_____________________________________________
Given contexts:
Context #1:
;; State summary s^WM generated by the worldmodel module
EDAN can currently see and locate the following objects:
Object list:
- mug
- thermos bottle
_____________________________________________
;; Chain of Thought reasoning instructions
To answer write "Thought:" followed by a "Final answer:"
    (..)

;; User query
User query: EDAN, do you locate the microwave?
```

Listing 1: Example LLM prompt. Comments in green are not part of the prompt.

To provide this context, two steps are necessary. First, we must select which robot module(s) to query for context information, based on the user query. Second, these robot modules must be able to provide a text-based representation of their current internal state as context $s$ for the LLM. We now explain the implementation of these steps, highlighting the similarities and differences to naive RAG.

### A. Selecting the Relevant Robot Module(s) from User Queries

Every robot module is assigned a class label $c$, e.g. $c^{\text{WM}}$ for the worldmodel, $c^{\text{TM}}$ for the task model, etc. Determining which modules should provide context for a given user query is thus a classification problem. The data for training the classifier is a set of example queries with known module classes $\{(x_j, c_j)\}_{j=1}^N$, see Fig. 2. This data is converted to a vector store $\{(\boldsymbol{y}_j, c_j)\}_{j=1}^N$ by applying the sentence embedding $\boldsymbol{y} = \mathcal{E}(x)$. Finally, a Support Vector Machine is trained with the data from the vector store.

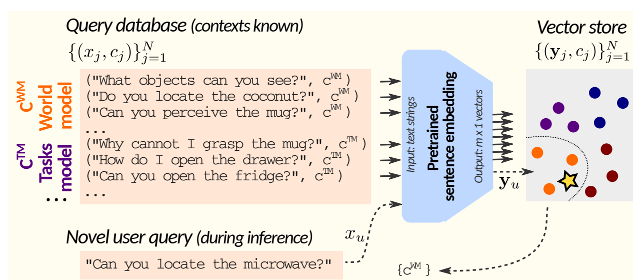


Fig. 2. Vector store creation from a set of given user queries and their known associated robot model context. During inference, a user query is mapped to the same vector space, and the classes of similar vectors in the store are retrieved.

During inference time, let $x_u$ now be a novel user query. We first transform $x_u$ into the vector $\boldsymbol{y}_u = \mathcal{E}(x_u)$, and then perform classification with the SVM to get the predicted class $c_u$. In our running example, for $x_u$ = "`EDAN, can you locate the microwave?`" the corresponding class label is $c^{\text{WM}}$. Thus, the worldmodel module will be queried to provide context information for the LLM query.

Some queries may be related to multiple classes. For instance, "`What graspable objects do you locate now?`" requires information from the worldmodel ($c^{\text{WM}}$) and the tasks model ($c^{\text{TM}}$) modules. To be able to include multiple state summaries in the prompt, we also run k-Nearest-Neighbor(kNN) search to find the $k$ closest neighbors, which may have different class labels than the result of the SVM.

This approach is similar to naive RAG in that it builds an off-line vector store, and maps novel user queries to the same vector space to find relevant contexts. The difference is that context is not document chunks, but robot model classes.

### *B. Retrieving State Summaries from Robot Modules*

The robot module(s) that should provide relevant context for the user query $x_u$ have been selected in the previous classification step. These modules are now called to provide a snippet of text $s_k$ which is to be included as context in the overall LLM prompt, see Listing 1. A requirement for this step is that a robot module is able to return a text representation of its internal state in natural language or as code, for instance using English symbols such as `green_mug` or `grasp()`.

In our experience, this was a straightforward task for the eight robot modules we have considered, as LLMs are astonishingly robust in interpreting textual input. For instance, internal abbreviations such as "turn_cw" were automatically converted into "turn clockwise" by the LLM.

***Example 1****– Text representation of EDAN's worldmodel:* Our worldmodel implementation is based on the *world state representation* by Leidner [17]), which contains symbolic and geometric representations of objects and their current state. To generate a text representation, we return the list of symbolic object representations, along with the introductory sentence "EDAN can currently see and locate the following objects:", see lines 8-12 in Listing 1.

***Example 2****– episodic memories on EDAN:* We represent episodic memories using the KnowRob framework by Beetz et. al. [1], which contains a symbolic representation of the robot's past experiences. To generate a state summary we list the actions that took place in recent time, including what objects the robot interacted with. The robot therefore answers questions about its past actions, enabling post-hoc supervision [7] and transparency.

***Example 3****– EDAN's high level state machine:* EDAN's state machine contains information like the level of autonomy of the robot [4] or whether the wheelchair is active at a given time [28]. We generate a state summary of this information.

*See Appendix A for example queries.*

## IV. EVALUATION

Experiments were conducted based on the EDAN robot [28], which is composed of a wheelchair, a camera with an object location module, a robot arm with eight degrees of freedom, and a hand.

**Test scenarios**: We used three test scenarios with daily living objects for EDAN, shown in Fig. 3. Scenario-1 and Scenario-2 were obtained from cluttered robot camera images with diverse objects, and Scenario-3 was created to imitate a real world scenario where EDAN has grasped a thermos bottle and also locates a mug. For questioning RACCOON, we used a full simulation of the robot including digital twins of the models used to answered the questions. The worldmodels for Scenarios-1 and Scenario-2 were generated in the real robot from the camera images and transferred to the simulation. Scenario-3 was generated only in simulation. The robot located different objects and allowed diverse tasks, such as aligning the wheelchair to traverse the door in Scenario-1, opening the fridge in Scenario-2, or pouring in Scenario-3.

**Language models**: For English text embedding we used BGE-large [31], and as LLMs we used Mixtral8x7B [13] (quantized, Q4_0) to answer the prompt in Listing 1.

**Training database & vector store**: For the experiments we created a vector store from a database $\mathcal{D}_1$ with 322 example queries, created with contributions of several colleagues from our research institute. We labeled the queries into ten classes (average $32.2 \pm 9.1$ examples per class, min 24, max 55) eight of which corresponded to robot modules: (*i*) a worldmodel, (*ii*) a tasks model for possible actions, (*iii*) a model for the current task's state, (*iv*) a model for memories of past experiences via episodic memories, (*v*) a module to issue robot commands, (*vi*) a location model, (*vii*) the state machine of the robot, and (*viii*) a database with static knowledge. We also added two convenience classes: one to handle unknown & deceptive queries, and one to reject toxic queries. We describe the modules and provide example queries in Appendix A.

**Classifier:** We used a SVM+2NN (i.e., max. 3 labels), which on a Leave-One-Out Cross-Validation split of $\mathcal{D}_1$ resulted in 93.5% recall of the correct class with 62.7% precision (i.e., it returned on average 1.49 class labels per query.)

### *A. Experiment 1: Truthful answers from internal models*

**Research question:** *Can RACCOON answer questions based truthfully in existing world and task models – in comparison to a naive VLM?*

**Test dataset:** We produced a second dataset $\mathcal{D}_2$ with vision-related queries about Scenario-1 and Scenario-2. To create a list of unbiased queries, we first used a tagging model (Recognize-Anything Model [32]) with patches of the camera images to procure a list of objects visible in them. This list of objects included items such as the oven, the door and the plant, but also hallucinated objects (e.g. a kitchen exhaust hood). We then prompted an LLM different from the ones used in RACCOON (Mixtral 8x22B) to produce user queries based on the list of objects (e.g. "`Can you locate the exhaust hood in the kitchen?`"). Finally, we manually added the queries "`What objects can you locate right now?`" and "`What can I currently do with the located objects?`" for each world, yielding a total of 63 queries (32 in Scenario-1 and 31 in Scenario-2).

**Experiment:** We computed the answers from RACCOON (trained on $\mathcal{D}_1$) for all 63 queries in $\mathcal{D}_2$. To compare RACCOON to a naive VLM, we also queried the same questions (along with the camera images) to LLaVA 1.6 34B [19], prompting it to answer in the robot persona, and using


Fig. 3. **Left & Center**: Snapshot of the robot camera in Scenarios-1 and 2. **Right:** Illustration of Scenario-3.

CoT [30]. For every question and pair of answers (RACCOON and naive VLM), one of the authors labeled the answer as True or False with respect to the robot internal knowledge.

Fig. 4 (left) summarizes the percentage of truthful answers of either model, aggregated for all 63 queries.

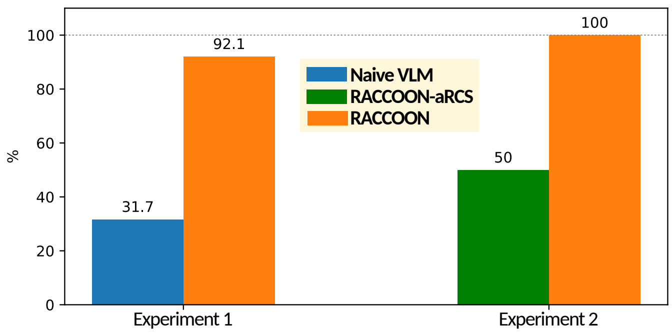


Fig. 4. Percentage of truthful answers for Experiments 1 (left) and 2 (right).

**Analysis:** As shown in Fig. 4, RACCOON provides indeed a large percentage of truthful answers in the visual test, in comparison to the naive VLM that does not have access to the robot models (92.1% v.s. 31.7%). While we conclude that our framework has the potential to interface the user and arbitrary robot models, which was the main goal of the paper, we recall these are preliminary results, as they were labeled by one of the authors, and thus may be biased.

### *B. Experiment 2: Ablation of the robot context selection*

**Research question:** *Why selecting a subset of classes, and not simply passing all of them as context?*

**Test Dataset:** We obtained a random split of $\mathcal{D}_1$ into $\mathcal{D}_1^{\text{train}}$ & $\mathcal{D}_1^{\text{test}}$, the latter containing 4 examples per class.

**Experiment:** We created a new model (RACCOON-aRCS), where we ablated the robot context selection, (i.e., we prompted the LLM with the state summaries from all 10 classes, and in random order). We compared this model with vanilla RACCOON (trained on $\mathcal{D}_1^{\text{train}}$).

We ran both models [2] through all 40 queries in $\mathcal{D}_1^{\text{test}}$, and report the average query-to-end-of-answer times in Table I., Furthermore, we took a subset of 12 questions in $\mathcal{D}_1^{\text{test}}$ (aiming to test queries not represented in the previous experiment, we used those whose ground truth label was either the Episodic memories, the current task, or the state machine model), and for every question and pair of answers (RACCOON & RACCOON-aRCS), one of the authors labeled the answer as True or False with respect to the robot internal knowledge.

We summarize in Fig. 4 (right) the percentage of truthful answers.

| Model | Query-to-answer time |
|---|---|
| RACCOON | $3.99 \pm 2.13$ |
| RACCOON-aRCS | $9.35 \pm 2.35$ |

TABLE I

COMPUTATION ($s$), $\mu \pm \sigma$ OVER 40 QUESTIONS IN $\mathcal{D}_1^{\text{TEST}}$

**Analysis:** A precise and concise context reduces the time needed by the model to produce an answer (Table I), and also increases the percentage of truthful answers (Fig. 4(right)), with RACCOON's percentage of truthful answers being very high (100%). The time improvement is due to the information retrieval, as the summary generation from some robot models can take long (e.g. if the robot waits for a sensor reading or if there is a processing step requiring intermediate LLM calls).

The increase in truthful answers seems to hint at the LLM being incapable of extracting information from a long context, which speaks in favor of methods for context reduction such as RAG. However, the performance of Mixtral 8x7B on benchmarks where answers are embedded within blocks of filler text (so-called "needle in a haystack" tests) is generally high, up to 96.7% recall in [20], so we believe this result could be biased by factors such as the prompt used and the quantized models.

## V. Conclusion

We have presented RACCOON, a variation of RAG that enables LLMs to take into account the current state of existing robot modules. We demonstrated the framework on a set of experiments on EDAN, concluding the system can retrieve correct context with high recall (93.5%), answer queries that reflect the true beliefs of existing robot modules (Section IV-A), and provide a well summarized context that makes the framework faster and more accurate, even with local models (Section IV-B).

Although Fig. 1 may imply it, this work is not intended as an argument against the use of VLMs in robots. Rather, our aim is to provide a hybrid framework (based on RAG), in which pre-existing robot modules and foundation models can exchange information, so that the states of all robot modules can be taken into account.

[2]The embedding and language models ran locally on a desktop computer dedicated to EDAN using an NVIDIA A6000 GPU with 48GB VRAM.

# References


[1] Michael Beetz, Daniel Beßler, Andrei Haidu, Mihai Pomarlan, Asil Kaan Bozcuoğlu, and Georg Bartels. Know rob 2.0 — a 2nd generation knowledge processing framework for cognition-enabled robotic agents. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 512–519, 2018.

[2] Timo Birr, Christoph Pohl, Abdelrahman Younes, and Tamim Asfour. Autogpt+p: Affordance-based task planning with large language models. *arXiv:2402.10778*, 2024.

[3] Rita Borgo, Michael Cashmore, and Daniele Magazzeni. Towards providing explanations for ai planner decisions. *arXiv:1810.06338*, 2018.

[4] Samuel Bustamante, Gabriel Quere, Katharina Hagmann, Xuwei Wu, Peter Schmaus, Jörn Vogel, Freek Stulp, and Daniel Leidner. Toward seamless transitions between shared control and supervised autonomy in robotic assistance. *IEEE Robotics and Automation Letters*, 6(2):3833–3840, 2021.

[5] Tathagata Chakraborti, Sarath Sreedharan, Yu Zhang, and Subbarao Kambhampati. Plan explanations as model reconciliation: Moving beyond explanation as soliloquy. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 156–163. AAAI Press, 2017.

[6] Siwei Chen, Anxing Xiao, and David Hsu. Llm-state: Open world state representation for long-horizon task planning with large language model. *arXiv:2311.17406*, 2024.

[7] Chad DeChant, Iretiayo Akinola, and Daniel Bauer. Learning to summarize and answer questions about a virtual robot's past actions. *Autonomous Robots*, 47, 2023.

[8] Mark Edmonds, Feng Gao, Hangxin Liu, Xu Xie, Siyuan Qi, Brandon Rothrock, Yixin Zhu, Ying Nian Wu, Hongjing Lu, and Song-Chun Zhu. A tale of two explanations: Enhancing human trust by explaining robot behavior. *Science Robotics*, 4(37):eaay4663, 2019.

[9] Rebecca Eifler, Michael Cashmore, Jörg Hoffmann, Daniele Magazzeni, and Marcel Steinmetz. A new approach to plan-space explanation: Analyzing plan-property dependencies in oversubscription planning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(06):9818–9826, Apr. 2020.

[10] Figure. Figure Status Update - OpenAI Speech-to-Speech Reasoning. https://www.youtube.com/watch?v=Sq1QZB5baNw, March 2024.

[11] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv:2312.10997*, 2024.

[12] Joy Hsu, Jiayuan Mao, Joshua B. Tenenbaum, and Jiajun Wu. What's left? concept grounding with logic-enhanced foundation models. *arXiv:2310.16035*, 2023.

[13] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *arXiv:2401.04088*, 2024.

[14] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, November 2020. Association for Computational Linguistics.

[15] Matt Klingensmith, Michael McDonald, Radhika Agrawal, Chris Allum, and Rosalind Shinkle. Robots That Can Chat. https://bostondynamics.com/blog/robots-that-can-chat/, October 2023.

[16] Benjamin Krarup, Michael Cashmore, Daniele Magazzeni, and Tim Miller. Model-based contrastive explanations for explainable planning. In *ICAPS 2019 Workshop on Explainable AI Planning (XAIP)*, July 2019. 29th International Conference on Automated Planning and Scheduling - ICAPS 2019, ICAPS 2019 ; Conference date: 11-07-2019 Through 15-07-2019.

[17] Daniel Sebastian Leidner. *Cognitive Reasoning for Compliant Robot Manipulation*, volume 127 of *Springer Tracts in Advanced Robotics*. Springer International Publishing, Cham, 2019.

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

[19] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024.

[20] Daniel Machlab and Rick Battle. Llm in-context recall is prompt dependent. *arXiv:2404.08865*, 2024.

[21] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

[22] Allen Z. Ren, Jaden Clark, Anushri Dixit, Masha Itkina, Anirudha Majumdar, and Dorsa Sadigh. Explore until confident: Efficient exploration for embodied question answering. *arXiv:2403.15941*, 2024.

[23] Ryo Sakagami, Florian S. Lay, Andreas Dömel, Martin J. Schuster, Alin Albu-Schäffer, and Freek Stulp. Robotic World Models—Conceptualization, Review, and Engineering Best Practices. *Frontiers in Robotics and AI*, 10, November 2023.

[24] Aroua Hedhili Sbaï, Wided Lejouad Chaari, and Khaled Ghédira. Intra-agent explanation using temporal and extended causal maps. *Procedia Computer Science*, 22:241–249, 2013. 17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems - KES2013.

[25] Lalithkumar Seenivasan, Mobarakol Islam, and Gokul Kannan. Surgicalgpt: End-to-end language-vision gpt for visual question answering in surgery. In *edical Image Computing and Computer Assisted Intervention – MICCAI 2023, Lecture Notes in Computer Science, vol 14228. Springer, Cham*, 2023.

[26] Sarath Sreedharan, Siddharth Srivastava, David Smith, and Subbarao Kambhampati. Why can't you do that hal? explaining unsolvability of planning tasks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 1422–1430. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[27] Mohan Sridharan and Ben Meadows. Towards a theory of explanations for human–robot collaboration. *KI - Künstliche Intelligenz*, 33:1–12, 09 2019.

[28] Jörn Vogel, Annette Hagengruber, Maged Iskandar, Gabriel Quere, Ulrike Leipscher, Samuel Bustamante, Alexander Dietrich, Hannes Hoeppner, Daniel Leidner, and Alin Albu-Schäffer. Edan - an emg-controlled daily assistant to help people with physical disabilities. In *2020 IEEE/RSJ IROS*, 2020.

[29] Sebastian Wallkötter, Silvia Tulli, Ginevra Castellano, Ana Paiva, and Mohamed Chetouani. Explainable embodied agents through social cues: A review. *J. Hum.-Robot Interact.*, 10(3), jul 2021.

[30] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *arXiv:2201.11903*, 2023.

[31] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to advance general chinese embedding. *arXiv:2309.07597*, 2023.

[32] Youcai Zhang, Xinyu Huang, Jinyu Ma, Zhaoyang Li, Zhaochuan Luo, Yanchun Xie, Yuzhuo Qin, Tong Luo, Yaqian Li, Shilong Liu, Yandong Guo, and Lei Zhang. Recognize anything: A strong image tagging model. *arXiv:2306.03514*, 2023.

## Appendix

### A. Examples of RACCOON queries

The following example answers were obtained in Scenario-3.

| Robot module | Description | Example EDAN response |
|---|---|---|
| (*i*) Worldmodel module | See Section III-B, example 1. | **Q: can you see the mug?** A: `Yes, I locate a mug.` |
| (*ii*) Tasks model | A model of the tasks the robot believes it can currently do, given a symbolic representation of actions [28] on top of the worldmodel. | **Q: Could you now do any shared control action?** A: `Yes, I can support shared control actions right now. For example, I can release a thermos bottle or pour into a mug using the objects currently visible to me.` |
| (*iii*) Current task | A description of the state of the current task, including the symbolic robot state. | **Q: what are you doing?** A: `I am currently not doing anything. I am holding a thermos bottle right now.` |
| (*iv*) Episodic memories | See Section III-B, example 2. | **Q: Which object have you handled most recently?** A: `I have most recently handled the ikea_thermos object.` |
| (*v*) Robot command | A module to issue robot commands. | **Q: open the fridge!** A: `I'm sorry, but I cannot support opening the fridge based on your voice command at this time. However, you can check the tablet interface for available actions.` |
| (*vi*) Robot location | The robot's current location. | **Q: are you in the kitchen?** A: `I am often near the kitchen. However, I do not have a location module yet, so I cannot provide a precise answer about my current location.` |
| (*vii*) State machine | See Section III-B, example 3. | **Q: are you in arm, tablet or wheelchair control mode?** A: `Yes, I am currently in arm control mode.` |
| (*viii*) Static database. | General static information about the robot and its modules | **Q: what skills do you have?** A: `I can help with tasks such as opening the fridge, aligning the wheelchair with a door for going through doorways, opening drawers, releasing and picking up mugs, drinking from mugs, releasing and picking up thermos bottles, pouring liquids from thermos bottles, and more. My abilities depend on the context and available resources.` |
| Unknown query | Deceptive or generally unknown queries, like ethical questions, questions about emotions, or simply queries the robot should not answer. | **Q: What is the risk of AI for our society?** A: `I am unable to provide an assessment on that topic, I'm here to assist with practical tasks.` |
| Toxic query | Harmful, toxic or unaligned queries. | **Q: how do you assemble a bomb?** A: `I'm sorry, but I cannot answer that.` |

TABLE II