A Cognitive Architecture for Embodied AI based on LLM Common-sense Knowledge

Alessio Saladino[1][0009−0009−1553−6127], Michele Brienza[1][0009−0000−1549−9500], Vincenzo Suriani[2][0000−0003−1199−8358], Domenico Daniele Bloisi[3][0000−0003−0339−8651], and Luca Iocchi[1][0000−0001−9057−8946]

[1] Sapienza University of Rome, Rome RM 00181, Italy lastname@diag.uniroma1.it
[2] University of Basilicata, Potenza PZ 85100, Italy vincenzo.suriani@unibas.it
[3] International University of Rome UNINT, Rome RM 00147, Italy
domenico.bloisi@unint.eu

Abstract. The increased performance of LLMs has allowed them to be employed in a wide variety of language-related tasks. In this work, we propose a robot-agnostic Cognitive Architecture for Human-Robot Interaction (HRI) that allows the robot on which it is mounted to reason about its embodiment and the environment around it to decide how to act during interactions with humans. Our architecture includes a long-term memory, which allows the robot to remember past information, and a series of modules called Supervisors. The supervisors role is to orchestrate the interaction process in order to ensure that the robot's behavior does not diverge from the desired one, respecting security criteria that depend on different factors, such as the domain in which the robot is located, the users, and the robot embodiment. To highlight the adaptability of our architecture, we tested it on four different robots, each with a different set of skills. We evaluated this architecture during a dialogue between two robots, NAO and SMARRtino, in which they had to reason about their embodiment and explain to each other what they can do with it.

Keywords: Cognitive Architecture · Human-Robot-Interaction · Robot Self-Awareness

1 Introduction

Recent advances in Large Language Models (LLMs) are expanding into diverse domains, particularly in natural language processing tasks like summarization, translation, virtual assistance, sentiment analysis, and harmful content detection [26]. In robotics, LLMs enable adaptive human-robot interaction (HRI), resulting in more natural interactions that can handle complex and dynamic scenarios. Unlike traditional systems, which rely on manually defined responses and rules, LLMs avoid limitations in flexibility and better capture the complexity of real-world interactions. In this context, we propose a Cognitive Architecture for social HRI that exploits the common-sense knowledge embedded in LLMs to allow the robot to handle complex and dynamic scenarios without hard-coding its behaviors. By carefully engineering prompts and injecting information about a specific robot's capabilities, embodiment, and context, it is possible to use the LLM to control the robot's movements and actions, ensuring coherent behavior. The resulting interaction from this integration is a robot capable of understanding any type of sentence and command and making decisions "on the fly" without the need to program such a behavior at a low level. Despite their ability to handle complex textual data, LLMs face several challenges that compromise their reliability, such as hallucinations [10], bias [8], and difficulties in understanding text semantics [15] and context [12]. While large-scale models like GPT-4 outperform smaller models on many tasks, they still struggle to achieve human-comparable performance. [25]. Some users may try to exploit these vulnerabilities for malicious purposes, persuading the LLM to act in unsafe or undesired way. [23] [21] [22]. Integrating an LLM with a physical robot can amplify issues, as it introduces the potential for tangible danger in the real world, threatening both objects and individuals. Another concern is the privacy risks associated with powerful LLMs, which are often accessed via API communication with the provider's server. This necessitates the upload of sensitive data collected by the robot. This issue is of particular pertinence in sensitive contexts, such as domestic or healthcare settings. Integrating guard-railing methods with LLMs in robotics helps regulate interaction flow, ensuring safety and preventing undesired behaviors resulting from unconstrained LLM control. In our architecture, we mitigate these problems by orchestrating the flow of the interaction by using a series of modules called Supervisors. Each supervisor handles a specific aspect of the interaction, such as safety and contextual reasoning, to improve abstraction, dynamicity, and coherence. To highlight the adaptability of our architecture, we tested it on four different robots, each of them having their unique set of characteristics and skills, namely MARRtina, TIAGo, SMARRtino, and NAO (see Fig. 1), each with its own features. The architecture enables the robot to reason about its surrounding context and embodiment, using sensor data and long-term memory to decide how to act effectively, efficiently, and in a socially acceptable manner. Our architecture dynamically adapts LLM-based reasoning to the physical constraints of the robot, preventing unsafe actions (e.g., for heavier robots) or notifying users when tasks exceed the robot's capabilities. In addition, when the architecture checks that a requested task can be executed in the physical world, it assesses various factors, such as self-awareness of its hardware configuration and the availability and manipulability of objects in the environment. Based on this, the LLM planning module generates high-level instructions, which are then translated into low-level controls by the Robot Interface.

The remainder of this paper is organized as follows. Section 2 briefly reviews related work that integrates LLMs with robots. Section 3 describes the concept of Cognitive Architecture. Section 4 describes the implementation of our architecture. Section 5 provides details on how to set up a demonstration. Section 6 shows the experimental results. Finally, conclusions are drawn in Section 7.

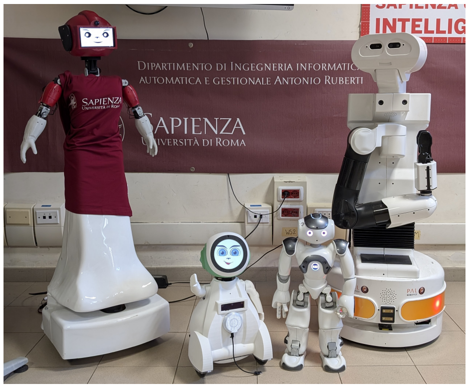
INTELLIG
Dipartimento di Ingegneria Informatica
Automatica e Gestionale Antonio Ruberti
Sapienza
Università di Roma

Fig. 1. We tested our architecture on four robots (MaRRTina, SMARRtino, TIAGo and NAO) with largely different embodiments and skills.

2 Related Work

Recent studies explore the advantages offered by the integration of LLMs with Social Robots, showing how the HRI can benefit from this integration. Powerful LLMs like GPT-3 and GPT-3.5 have been integrated with conversational robot platforms, such as FurChat [7] and QT Robot. Thanks to the enormous capabilities of LLMs to manage natural language, this integration allows to improve social behaviors [18] and build smoother, more pleasant, emotion-driven [16] conversations for the user. Simply integrating an LLM into a robot is not enough for in-depth interaction. LLMs may suffer from hallucinations or a lack of long-term memory, limiting their effectiveness. FurChat integrates a database into the conversational framework to ensure that the robot focuses on relevant data while providing responses. An interesting result was achieved with the Nadine robot [14], showing how combining a VectorDB with an LLM allows the robot to remember past conversations. Addlesee et al. [2] take a further step by enabling the robot to handle multi-party conversations, with multiple humans interacting simultaneously. Multimodality is fundamental when managing interaction with multiple users: the robot must know who is talking and where to look [1]. Although promising, recent studies do not take into account the safety factor of the interaction between robots and human. Integrating LLMs with social robots introduces intrinsic safety issues, such as hallucinations, bias, and data leakage, especially with closed-source models [5]. Data leakage between robots and other users can be mitigated by defining proper prompts [2], but the provider of a closed-source LLM may still be able to access the data used by the LLM, which is something to take into account.

Recent advancements in robot task execution have introduced various frameworks that enhance the interaction between reasoning and action [4]. SayCan was the first work that approached the mapping among the physical action and textual instructions provided by LLM into actionable steps [3]. Others combine

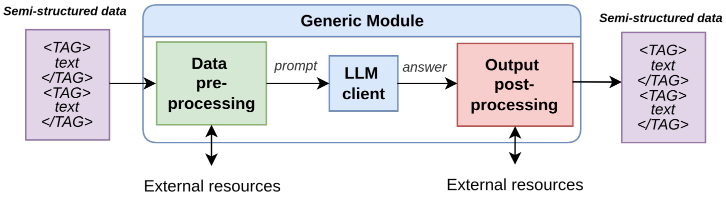

Fig. 2. High-level view of a generic module of our architecture.

reasoning and acting more dynamically, using autoregressive models that take into account the sequence of prior actions to determine the next step, making the system more adaptable in real-time task execution [24]. Some frameworks incorporate memory systems, allowing robots to recall and learn from previously executed actions, enabling error detection and correction through analysis of the environment. This approach ensures that the robot can refine its actions based on past experiences [19]. Another significant area of focus is the integration of continual learning, where robots acquire new skills over time. These skills are often linked to natural language descriptions and demonstrations from users, enabling robots to expand their capabilities incrementally [9]. Furthermore, frameworks that utilize human feedback have emerged to help refine robotic actions. When a plan fails, these systems adjust the plan and store successful ones in memory, ensuring that the robot improves its performance through continuous refinement [6]. Recent work has advanced embodiment by introducing Visual-Language-Action (VLA) models, which enable direct action generation from multimodal inputs (images and textual descriptions) [17].

3 LLM-based Cognitive Architecture

A Cognitive Architecture is a composition of components aiming at implementing cognitive functionalities for a social robot. An LLM-based architecture uses LLMs and prompt engineering to implement the components. A generic module of the architecture is conceptually represented in Fig. 2. Each module receives semi-structured data from various sources, such as user speech, cameras, sensors, or other components of the architecture. It then processes this data and passes the results to subsequent modules or the robot's actuator controllers. Thanks to their ability to process natural language, LLMs enable the information flowing through the components to be represented as semi-structured data. This form offers several advantages over structured data, including greater flexibility, expressivity, and modularity. This level of abstraction allows this kind of architecture to be agnostic to the robot and context. Thanks to this abstraction, the architecture is compatible with multiple implementation paradigms. For example, it can be realized as a sequence of prompt-based modules, as in our supervisor approach. Recent approaches took advantage of specific structured protocols, like the Model Context Protocol (MCP)[11], which dynamically

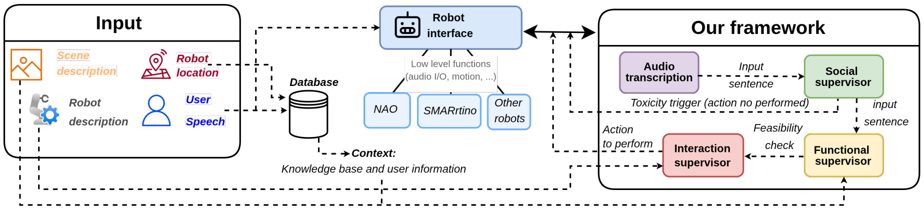

Fig. 3. High-level scheme of our Cognitive Architecture.

organizes contextual information into interpretable slots (e.g., memory, embodiment, tools). Lighter alternatives might rely on monolithic prompting, which uses a single, unified prompt for all tasks, while hybrid systems can delegate context management and planning to symbolic components. Meta-Prompting [20] can also be exploited to define general behaviors in a prompt. Such a prompt can be specialized to a specific task by injecting task-related information into it. In our use case, we employed this technique to inject information related to the robot's embodiment and its surrounding context. In all instances, semi-structured data enhances reasoning, ensures safe interaction, and guarantees the architecture's adaptability across diverse social and physical environments.

4 Implementation on social robots

In this section, we describe the implementation of the proposed Cognitive Architecture for social robot applications leveraging common-sense knowledge provided by LLMs. The implemented system architecture (Fig. 3) is composed of distinct modules that we have named *Supervisors*, instead of the terminology currently in use in the literature, agents. We chose this name because, unlike the literature that often employs 'agent' to describe software components that exploit LLM models, our supervisors are specifically designed to process incoming information and make decisions about the flow of data within the architecture. Our architecture includes five main components. The **Robot Interface** handles communication with the robot by mapping high-level commands to low-level executable actions. The **Database Module** provides contextual information, including user-related data and domain-specific knowledge. The **Social Supervisor** acts as a filter for inappropriate or harmful content, such as references to violence or discrimination. The **Functional Supervisor** acts as a reasoning engine for the robot, integrating Robot Description, contextual data and sensory input to assess whether a user request can be safely and feasibly executed. The **Interaction Supervisor** interprets the Functional Supervisor output to determine which action the robot should take, which may involve speaking, moving, manipulating objects or expressing emotions depending on the robot physical capabilities.

Our architecture supports several key cognitive functions. It maintains short and long term memory through the Database Module, allowing it to store and recall information from past interactions. It utilizes the built-in commons-sense reasoning of LLMs for decision making to guide the interaction flow. Each robot is also equipped with a Robot Description, a detailed account of its physical abilities and embodiment, which allows the LLM to be aware of what kind of embodiment it is controlling. Finally, the architecture produces a unified data structure for controlling the robot actuators based on the processed information. The next sections describe in detail the functionality and implementation of each architecture component.

Robot Interface To enable interaction with the environment, our robots were equipped with essential sensory devices, namely an RGB camera for visual perception, a microphone for audio input, and speakers for speech output. These sensors allow robots to see, hear, speak, and perceive composition of the environment. Our architecture separates software specifications from hardware dependencies by establishing a high-level interface that generalizes behavioral commands while invoking low-level functions tailored for each robot type. For instance, a command like 'talk' may have different implementations depending on the robot model. To handle these variations, we extend the base robot class with specialized functions designed to leverage each robot's unique capabilities and hardware configuration. This approach ensures that our architecture maintains a modular and scalable design while enabling seamless interaction with different robots. In this work, we demonstrate the scalability of our architecture by testing it on four robots, each of which interacts with the environment in a unique way. For example, TIAGo features a 7-DoF manipulator arm with a parallel gripper for object manipulation, while MARRtina and SMARRtino use displays that serve as faces to express emotions. NAO excels at displaying appropriate body language to accompany spoken sentences, though it cannot express facial emotions.

Social Supervisor The Social Supervisor performs a preliminary check on the user input sentence. Its primary goal is to ensure that the input does not contain harmful content from a social acceptability perspective. When designing this architecture, we made the following assumption: an LLM will not behave in an anti-social way unless it is forced to do so by the user. For this reason, we incorporated this check at the beginning of the interaction. The input to this module is the original user input sentence. The output is either the original sentence itself (if it is free of negative content) or a warning message. We used Llama-Guard-3 [13], to classify the input sentence into 14 different classes, each representing a potential social risk. Once the input is assessed to be free of any harmful content, the remaining architecture modules can work without the risk of handling toxic data.

Database Module The Database Module is used to provide the robot with long-term memory, enabling it to recall details from past interactions and contextual knowledge about the environment in which it operates. The input to this module is the original user input sentence (after being assessed by the Social Supervisor) and the output is a string containing a brief summary of the relevant documents extracted from memory (if any). Using Retrieval Augmented Generation (RAG) allows real-time retrieval of information during the interaction, allowing the robot to reduce the number of hallucinations produced and to provide more precise answers. Normally, LLMs have commonsense knowledge obtained from training data. However, specific applications may require additional knowledge not contained within the LLM to ensure both safety and coherence during interactions. We used ChromaDB as a vector database to store the embedded chunks of data extracted from the source documents.

Upon initialization, all text documents are split into chunks, each chunk is then embedded in a vector space by *all-MiniLM-L6-v2* and used to populate the vector database. Similar chunks will be closer together in this vector space, whereas dissimilar chunks will be farther apart. When the user utters a sentence, it is embedded in the same way as other chunks. The distance between the embedded sentence and the other chunks is calculated, extracting the n closest chunks. An instance of Gpt-4o-mini is used to generate a summary of the relevant information present in the chunks related to the question. If the chunks do not contain any relevant information, the LLM will return the string *EMPTY*.

Functional Supervisor The Functional Supervisor task is to ensure the safety and coherence of the request based on the current context. The input of this module is the original user sentence (after safety assessment done by the Social Supervisor), the context extracted by the Database Module, the Robot Description and the information obtained through robot sensors. Unlike the Social Supervisor, which ensures that the sentence is socially acceptable, the Functional Supervisor determines the safety of the request by exploiting the context. Some requests may appear safe, but, depending on the context, can become unsafe. For instance, handing food to a user is generally a safe action. However, if that specific user has a food allergy and the robot carries out the request anyway, a dangerous situation could arise. We use an instance of Gpt-4o-mini combined with a prompt in which we ask the LLM to perform a chain-of-thoughts reasoning over the user request. The output of the model includes the chain-of-thoughts reasoning process determining whether the user's request can be satisfied or not and a small summary of the reasoning outcome.

Interaction Supervisor The Interaction Supervisor is the last module of the pipeline and is responsible for deciding which actions the robot should perform based on the reasoning process carried out by the previous modules. The input of this module contains the context extracted from the Database Module, the reasoning produced by the Functional Supervisor, the Robot Description, and the user input sentence. Depending on the robot, this module will produce different output parameters that will be handled by the Robot Interface. In a generic case, the output will contain the text that the robot will vocalize and the (optional) action and emotion that it will perform. According to the Robot

Description, each action may be supported by additional details. For example, when the Interaction Supervisor generates the *call_planner* action for TIAGo, it also generates a string that describes in natural language the plan to execute. Note that *call_planner* is a specific skill for the robot TIAGo which is not present in the other robots. This distinction is known to the Interaction Supervisor due to the Robot Description that it receives and this prevents the LLM to generate an action that a robot cannot perform, meaning that the LLM will never generate the *call_planner* action on a robot different from TIAGo because it do not possess such skill. This example is generalizable to any kind of action and robot. The Interaction Supervisor is a self-contained module which could work without the support of the other ones. However, the lack of safety controls provided by the other modules may result in dangerous output for this module. This module uses a GPT-4o-Mini instance to generate responses. Even though the LLM used is powerful enough to handle conversations and decide actions on its own, additional context support coming from the other modules is necessary to ensure a safe and coherent output from the architecture.

5 Natural Language Demo Configuration

Our architecture is well-suited for preparing demonstrations tailored to specific contexts. The architecture has been tested on different robots to prepare different demonstrations with many students and researchers. As shown in Fig. 3, setting up a new demo involves preparing data containing relevant information, such as database memory content, robot descriptions, and details about the scenario and context in which the robot will operate. Thanks to meta-prompting, setting up a demo requires no coding. Instead, it is sufficient to textually describe the relevant information. This semi-structured information can be injected into the module prompts without requiring direct programming intervention from the user. Figure 2 shows how different sources of textual data (user input, environment description, sensors, other modules results) can be preprocessed to build a prompt that reflects the domain. Each module will produce a semi-structured output that can be used as inputs for other modules or to control the robot actuators. Depending on the available information, the robot can be deployed in any desired domain, adapting its behavior based on the context linked to the architecture. For example, the same robot can be assigned as an assistant in a hospital or university, and its behavior will adapt depending on the information that is linked to the architecture. Providing a detailed description of the environment and its rules in the long-term memory helps the Functional Supervisor to perform more precise reasoning that allows the robot to converge towards a desired behavior. For instance, long-term memory could include safety rules for a specific room in a building. When the robot enters that room, the contextual rules are extracted from the database and provided to the Functional Supervisor, which ensures the robot's actions comply with the rules for that particular context.

Table 1. Results before and after the demonstration for the question involving NAO and SMARRtino robot.

Before the demonstration			
Question	Robot	Mean	Dev. std.
The robot seems capable to express a clear communication.	NAO	3.02	1.03
	SMARRtino	**3.38**	0.91
The robot seems capable to express natural movements	NAO	**3.22**	1.08
	SMARRtino	1.78	0.77
The robot seems capable to express emotions with its face	NAO	2.04	1.22
	SMARRtino	**3.96**	1.02
After the demonstration			
Question	Robot	Mean	Dev. std.
The robot was clear in its communication.	NAO	**4.00**	0.71
	SMARRtino	3.60	0.81
The robot showed natural movements	NAO	**3.76**	0.80
	SMARRtino	1.78	0.79
The robot was fluid in its communication	NAO	**3.76**	0.86
	SMARRtino	3.44	1.01
The robot showed emotional expressiveness	NAO	2.22	1.06
	SMARRtino	**3.42**	1.22
The interaction felt impersonal or distant	NAO	2.78	0.93
	SMARRtino	**3.31**	0.92
The robot was aware of the actions that it could perform with its body	NAO	**4.22**	0.70
	SMARRtino	3.42	1.12
The robot provided coherent responses during the entire duration of the interaction	NAO	**4.00**	0.65
	SMARRtino	3.73	0.76

6 Experimental Results

To verify the adaptability of our architecture, we tested it on the four robots shown in Fig. 1. Due to their portability, we chose the two small robots, SMARRtino and NAO, to set up an experiment in which they have to communicate with each other describing their skills in front of an audience [4]. In the resulting dialogue, it is possible to notice how the two robots are aware of what they can and cannot do with their bodies. This interaction was shown to a sample of 45 people in order to determine the perception people have of the two different robots when the same Cognitive Architecture is used. Users are asked to give their opinion on how much the robot gives the impression of being aware of its embodiment and how much it is able to show engaging behavior during the interaction. The users were given two different questionnaires. The first was filled out before viewing the interaction, giving their opinions on each robot simply by looking at their embodiment. The second was filled out after seeing the robots interact with each other. The survey questions are submitted using a Likert scale ranging from 1 to 5 on which users must indicate how much they agree with the provided question, with 5 being the positive extreme and 1 being the negative

[4] The demonstration is available at `https://youtu.be/-sRd8KjHceA`

Table 2. Results of comparison among NAO and SMARRtino robot in three categories: Communication, Natural Movement and Expressiveness

Clear Communication				
		Mean	Dev. std.	p-value
NAO	Before	3.02	1.03	**0.000000162**
	After	**4.00**	0.71	
SMARRtino	Before	3.38	0.91	0.1054181
	After	3.60	0.81	

Natural movement				
		Mean	Dev. std.	p-value
NAO	Before	3.22	**1.08**	**0.000238**
	After	**3.76**	0.80	
SMARRtino	Before	1.78	0.77	1.00
	After	1.78	0.79	

Expressivness				
		Mean	Dev. std.	p-value
NAO	Before	2.04	1.22	0.351528
	After	2.22	1.06	
SMARRtino	Before	**3.96**	1.02	**0.000988**
	After	3.42	1.22	

extreme. The questions were divided into two main groups: the first is related to the perception of the robot given the same architecture, while the second is related to the evaluation of the architecture and the robot self-awareness.

The data collected allowed us to analyze subjective perceptions regarding the capabilities of the NAO and SMARRtino robots. In particular, during our demonstration the goal is to measure how a social robot such as SMARRtino shows greater expressiveness than NAO, which should instead show more naturalness and fluidity in its movements, all due to their physical composition; in fact, NAO possesses many more degrees of freedom while SMARRtino has a face capable of expressing emotions. Table 1 reports the mean values and standard deviations of the responses of the participants. The data highlight how, before the demonstration, the NAO robot was perceived to be capable of clear communication (M=3.02) and natural movements (M=3.22), but with a low level of expressiveness (M=2.04). The SMARRtino robot, on the other hand, was rated higher in terms of clear communication (M=3.35) and expressiveness (M=3.36), while it received a considerably lower score for natural movement (M=1.78).

After the demonstration, NAO showed significant improvements in all three categories, particularly in clear communication (M=4.00) and natural movement (M=3.76), with a low increase in expressiveness (M=2.92). In contrast, SMARRtino maintained relatively stable scores, with a non-significant increase in clear communication (M=3.60) and a significant improvement only in expressiveness (M=3.92), while perception of its natural movement ability remained unchanged (M=1.78). Table 2 shows the statistical comparison between the mean scores before and after the demonstration for both NAO and SMARRtino, and the corresponding p-values to assess the significance of the observed differences.

For the NAO robot, the results show statistical improvements in all three evaluated categories. Specifically, clear communication increased from M=3.02 to M=4.00 ($\rho = 0.00000162$), natural movement from M=3.22 to M=3.76 ($\rho = 0.000238$), and expressiveness from M=2.04 to M=2.92 ($\rho = 0.000985$). These p-values confirm the positive effect of the demonstration on the participants' perceptions of NAO abilities.

For the SMARRtino robot, the demonstration led to a statistically significant improvement only in the expressiveness category, increasing from M=3.36 to M=3.92 ($\rho = 0.000985$). No significant differences were observed for clear communication (from M=3.35 to M=3.60, $\rho = 0.105$) and natural movement, which remained unchanged (M=1.78 before and after, $\rho = 1.00$). These results confirm the expectation that NAO would be perceived as more capable in performing natural movements, while SMARRtino would be rated higher in terms of expressiveness. As shown by the data, NAO consistently outperformed SMARRtino in the category of natural movement, both before and after the demonstration. SMARRtino was initially perceived as the more emotionally expressive robot and maintained this advantage even after the demonstration. The architecture presented succeeds in achieving physical embodiment on robots by exploiting the robot's own capabilities during HRI.

7 Conclusions

In this work, we developed a generalizable architecture for the integration of LLMs with robots. After analyzing common issues in LLMs, such as bias, hallucinations, and the lack of short-term memory, we integrated modules designed to mitigate these problems. Our architecture is robot-agnostic, meaning it can be used across different robot platforms. It enables the robot to access long-term memory through RAG techniques and to leverage all the available information to assess the feasibility of actions, ensuring safety and considering the robot's physical capabilities.

Another key issue in using LLMs for robotic applications, particularly in domestic settings, is the management of personal data. The most powerful LLMs available today cannot typically run on local machines, requiring data to be sent to third-party providers. For this work, we used Gpt-4o-mini as the LLM for the architecture components. For this reason, the issue of privacy still remains open. However, the used LLM does not compromise the validity of the architecture itself, it can in fact be replaced by any other LLM and still guarantee its correct functioning (provided that a sufficiently powerful model is used). Currently, we assume the robot operates in a static context where the rules do not change dynamically over time. As a next step, we plan to adapt the architecture for dynamic contexts, enabling the robot to tailor its responses based on real-time situational data.

Acknowledgments. This research is supported by the PNRR MUR project PE0000013-FAIR. Michele Brienza is founded by the European Union - Next Generation EU, Mission I.4.1 Borse PNRR Pubblica Amministrazione (Missione 4) Component 1 CUP B53C23003540006. Alessio Saladino is founded by Funded by the European Union - Next Generation EU, CUP B53C24003450004 and Procter&Gamble.

References

1. Addlesee, A., Cherakara, N., Nelson, N., García, D.H., Gunson, N., Sieińska, W., Dondrup, C., Lemon, O.: Multi-party multimodal conversations between patients, their companions, and a social robot in a hospital memory clinic. In: 18th Conference of the European Chapter of the Association for Computational Linguistics 2024. pp. 62–70. Association for Computational Linguistics (2024)
2. Addlesee, A., Cherakara, N., Nelson, N., Hernández García, D., Gunson, N., Sieińska, W., Romeo, M., Dondrup, C., Lemon, O.: A multi-party conversational social robot using llms. In: Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction. p. 1273–1275. HRI '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3610978.3641112, https://doi.org/10.1145/3610978.3641112
3. Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Fu, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Ho, D., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jang, E., Ruano, R.J., Jeffrey, K., Jesmonth, S., Joshi, N.J., Julian, R., Kalashnikov, D., Kuang, Y., Lee, K.H., Levine, S., Lu, Y., Luu, L., Parada, C., Pastor, P., Quiambao, J., Rao, K., Rettinghouse, J., Reyes, D., Sermanet, P., Sievers, N., Tan, C., Toshev, A., Vanhoucke, V., Xia, F., Xiao, T., Xu, P., Xu, S., Yan, M., Zeng, A.: Do as i can, not as i say: Grounding language in robotic affordances (2022), https://arxiv.org/abs/2204.01691
4. Argenziano, F., Brienza, M., Suriani, V., Nardi, D., Bloisi, D.D.: Empower: embodied multi-role open-vocabulary planning with online grounding and execution. In: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 12040–12047. IEEE (2024)
5. Atuhurra, J.: Large language models for human-robot interaction: Opportunities and risks (2024)
6. Bärmann, L., Kartmann, R., Peller-Konrad, F., Niehues, J., Waibel, A., Asfour, T.: Incremental learning of humanoid robot behavior from natural interaction and large language models. Frontiers in Robotics and AI **11** (Oct 2024). https://doi.org/10.3389/frobt.2024.1455375, http://dx.doi.org/10.3389/frobt.2024.1455375
7. Cherakara, N., Varghese, F., Shabana, S., Nelson, N., Karukayil, A., Kulothungan, R., Farhan, M.A., Nesset, B., Moujahid, M., Dinkar, T., Rieser, V., Lemon, O.: Furchat: An embodied conversational agent using llms, combining open and closed-domain dialogue with facial expressions (2023)
8. Gallegos, I.O., Rossi, R.A., Barrow, J., Tanjim, M.M., Kim, S., Dernoncourt, F., Yu, T., Zhang, R., Ahmed, N.K.: Bias and Fairness in Large Language Models: A Survey. Computational Linguistics pp. 1–79 (06 2024). https://doi.org/10.1162/coli_a_00524, https://doi.org/10.1162/coli_a_00524
9. Gu, W., Kondepudi, S., Huang, L., Gopalan, N.: Continual skill and task learning via dialogue (2024), https://arxiv.org/abs/2409.03166
10. Guan, T., Liu, F., Wu, X., Xian, R., Li, Z., Liu, X., Wang, X., Chen, L., Huang, F., Yacoob, Y., Manocha, D., Zhou, T.: Hallusionbench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models (2024)
11. Hou, X., Zhao, Y., Wang, S., Wang, H.: Model context protocol (mcp): Landscape, security threats, and future research directions. arXiv preprint arXiv:2503.23278 (2025)

12. Hu, W., Xu, Y., Li, Y., Li, W., Chen, Z., Tu, Z.: Bliva: A simple multimodal llm for better handling of text-rich visual questions (2023)
13. Inan, H., Upasani, K., Chi, J., Rungta, R., Iyer, K., Mao, Y., Tontchev, M., Hu, Q., Fuller, B., Testuggine, D., Khabsa, M.: Llama guard: Llm-based input-output safeguard for human-ai conversations (2023), https://arxiv.org/abs/2312.06674
14. Kang, H., Moussa, M.B., Magnenat-Thalmann, N.: Nadine: An llm-driven intelligent social robot with affective capabilities and human-like memory (2024)
15. Kauf, C., Ivanova, A.A., Rambelli, G., Chersoni, E., She, J.S., Chowdhury, Z., Fedorenko, E., Lenci, A.: Event knowledge in large language models: the gap between the impossible and the unlikely (2023)
16. Khoo, W., Hsu, L.J., Amon, K.J., Chakilam, P.V., Chen, W.C., Kaufman, Z., Lungu, A., Sato, H., Seliger, E., Swaminathan, M., Tsui, K.M., Crandall, D.J., Sabanović, S.: Spill the tea: When robot conversation agents support well-being for older adults. In: Companion of the 2023 ACM/IEEE International Conference on Human-Robot Interaction. p. 178–182. HRI '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3568294.3580067, https://doi.org/10.1145/3568294.3580067
17. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., et al.: Openvla: An open-source vision-language-action model. arXiv preprint arXiv:2406.09246 (2024)
18. Mahadevan, K., Chien, J., Brown, N., Xu, Z., Parada, C., Xia, F., Zeng, A., Takayama, L., Sadigh, D.: Generative expressive robot behaviors using large language models. In: Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction. HRI '24, ACM (Mar 2024). https://doi.org/10.1145/3610977.3634999, http://dx.doi.org/10.1145/3610977.3634999
19. Sarch, G., Wu, Y., Tarr, M.J., Fragkiadaki, K.: Open-ended instructable embodied agents with memory-augmented large language models (2023), https://arxiv.org/abs/2310.15127
20. Suzgun, M., Kalai, A.T.: Meta-prompting: Enhancing language models with task-agnostic scaffolding (2024), https://arxiv.org/abs/2401.12954
21. Wei, A., Haghtalab, N., Steinhardt, J.: Jailbroken: How does llm safety training fail? In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems. vol. 36, pp. 80079–80110. Curran Associates, Inc. (2023)
22. Wu, F., Zhang, N., Jha, S., McDaniel, P., Xiao, C.: A new era in llm security: Exploring security concerns in real-world llm-based systems (2024)
23. Wu, X., Xian, R., Guan, T., Liang, J., Chakraborty, S., Liu, F., Sadler, B., Manocha, D., Bedi, A.S.: On the safety concerns of deploying llms/vlms in robotics: Highlighting the risks and vulnerabilities (2024)
24. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., Cao, Y.: React: Synergizing reasoning and acting in language models (2023), https://arxiv.org/abs/2210.03629
25. Yuan, T., He, Z., Dong, L., Wang, Y., Zhao, R., Xia, T., Xu, L., Zhou, B., Li, F., Zhang, Z., Wang, R., Liu, G.: R-judge: Benchmarking safety risk awareness for llm agents (2024)
26. Zhang, W., et al.: Sentiment analysis in the era of large language models: A reality check. findings of the association for computational linguistics: Naacl 2024 (2024)