# SinfonIA Uniandes: Winning Team of the RoboCup@Home Social Standard Platform League 2024

David Cuevas[0009−0001−1391−953X] and Luccas Rojas[0009−0002−3792−1113]

[1] IMAGINE: Computación Visual, I+D+i, SinfonIA Uniandes
[2] Universidad de los Andes, 111711 Bogotá, Colombia
{sinfonia,d.cuevas}@uniandes.edu.co
https://sinfoniauniandes.github.io/SinfonIA-web/home,
https://www.uniandes.edu.co/es

**Abstract.** This work explores the improvements and methods that led to SinfonIA Uniandes' success in winning the RoboCup@Home Social Standard Platform League (SSPL) 2024, using the Pepper robot from SoftBank Robotics. Considering the challenges due to human-robot interaction (HRI) and the hardware constraints, our focus was on improving Pepper's ability to offer practical assistance in home tasks. Key improvements include a robust person-following system that combined YOLO-based visual detection with a PID controller, achieving 80% accuracy in tracking individuals during short walks (under one minute). To improve General Purpose Service Robot (GPSR) tasks and perception capabilities we integrated OpenAI's GPT-4o, significantly enhancing Pepper's interaction and functionality.
This enabled Pepper to do more complex operations like determining whether a person is wearing shoes or interpreting gestures with an 89% success rate in executing autogenerated GPSR tasks. The integration of GPT-4o with OpenAI's locally deployed Whisper system provided robust speech recognition, facilitating smoother and more natural conversations, crucial for complex human-robot interactions. Additionally, GPT-4o's multimodal capabilities enabled advanced image processing, enhancing the robot's understanding of its surroundings. Our work with Pepper underscores the potential of AI to address key hardware and perception challenges, empowering social robots to effectively assist humans in dynamic environments.

**Keywords:** RoboCup@Home · RoboCup · Social Standard Platform League · Pepper · SinfonIA Uniandes · YOLO · ChatGPT · Whisper · Human-Robot Interaction

## 1 Introduction

In the year 2024, SinfonIA Uniandes participated in the RoboCup@Home Social Standard Platform League that was held in Eindhoven and won. The team consisted of four undergraduate engineering students and one undergraduate law

student working together with a single objective: facing challenges posed during the competition.

This paper is organized as follows: Section 2 introduces the RoboCup@Home Social Standard Platform League competition and its challenges. Section 3 describes the Pepper robot and its hardware. Section 4 details the software architecture and key innovations. Section 5 discusses the current research. Lastly, Section 6 concludes the paper with a summary and outlook.

## 2 RoboCup@Home Social Standard Platform League

SinfonIA Uniandes has been an active participant in the **RoboCup@Home** Social Standard Platform League since 2019, always using the Pepper robot manufactured by SoftBank Robotics. This league evaluates robots in domestic and social environments, forcing teams to create software applications to facilitate everyday tasks. In 2024, our team participated in RoboCup in Eindhoven with a focus on advancing the development of social robotics that prioritize generalization over task-specific capabilities.

The Social Standard Platform League competition is designed in three stages. The first stage benchmarks core functionalities, including person tracking and following, speech understanding, recognition of objects, and navigation. Tasks included Carry My Luggage, where a robot helps a person transport their belongings, and Receptionist, where a robot greets and guides visitors. The rest of the tasks included grocery storage, breakfast setup preparations, and implementation of General Purpose Service Robot (GPSR) commands ranging from determining if a person had their shoes on to recognizing specific gestures.

During the second stage, all the core functionalities were combined in more involved tasks. The robots performed tasks like Clean the Table, which involved clearing objects from a table, and Restaurant, where they retrieved and served orders to customers in a simulated restaurant, detecting and reaching tables without prior guidance. The Enhanced General Purpose Service Robot (EGPSR) task required the robot to locate individuals in the arena and respond to autogenerated commands.

The final round featured the top-performing teams, with a focus on innovative applications of social robotics. SinfonIA Uniandes stood out for its advanced human-robot interaction capabilities and very efficient help solutions. Our participation in RoboCup demonstrates the potential of social robots to address practical problems while advancing both the research and applications of the robotics domain.

## 3 Hardware Setup

In alignment with the regulations established by the RoboCup@Home Social Standard Platform League, our hardware configuration adhered to the league's stipulations that disallow any alterations to the standardized robot platform. For this year's competition, SinfonIA Uniandes utilized a Pepper robot supplied

in the competition, necessitating a comprehensive reinstallation of our software on this device.

The Pepper model has fixed hardware, including a 3D camera, microphones, tactile sensors, and a tablet interface. The robot is functional but has serious limitations. The mechanical design is limited by a reduced range of motion, featuring weak arms with very restricted mobility. The fingers close all together, not one by one, which makes grasping tasks difficult. In addition, the maximum speed of Pepper is just 0.5 m/s, which is very slow compared to other robots.

The robot's sensors also have significant shortcomings. The depth camera only operates effectively from 40 centimeters in front of the robot and is both imprecise and low quality, making precise grasping tasks nearly impossible. Similarly, Pepper's regular cameras struggle with performance; higher resolutions cause significant slowdowns, forcing us to use very low resolutions. The laser sensors are insufficient for robust localization, adding to the difficulty of reliable navigation.

To overcome most of these challenges, the majority of computations were offloaded to an external computer using the ROS distributed node system. This setup handled resource-intensive tasks such as Whisper speech recognition, YOLO object detection, and face recognition. By leveraging external processing, we were able to extend the capabilities of the robot despite its hardware limitations.


**Fig. 1.** Pepper Robot. Source: SoftBank Robotics (https://corporate-internal-prod.aldebaran.com)

## 4 Methodology

**Architecture** Since we did not have root access on the Pepper robot, we could not install the required software (ROS) directly on its onboard computer. To overcome this limitation, we used Gentoo Prefix, which is a minimal user-space portage system. In addition, we have included the ros-overlay (based on ROS Kinetic), with additional changes to make Pepper's capabilities more complete by integrating the latest available software. The foundation of our system consists of the raw Pepper hard disk image version 2.5.5.5, which acts as a cross-compilation environment.

To integrate ROS into the Pepper robot, a proprietary driver was developed to replace the current naoqi driver. This new driver has two separate parts: a driver for basic functionalities written in C++ and an extended functionality driver built in Python. The C++ part is in charge of the basic functionalities such as navigation, perception, or speech processing, while the Python part is in charge of less intensive operations like displaying images on the robot's tablet. This separation brings better performance and modularity.

Together, both drivers provide low-level instructions for control, and modules like navigation, perception, speech, and manipulation handle higher-level tasks. At the highest level, the task module connects all of these elements to perform complex tasks, ensuring seamless interaction between the robot's capabilities and enabling advanced operations. The software architecture is visually represented in Figure 2, which illustrates how these modules interact and are organized within the system.

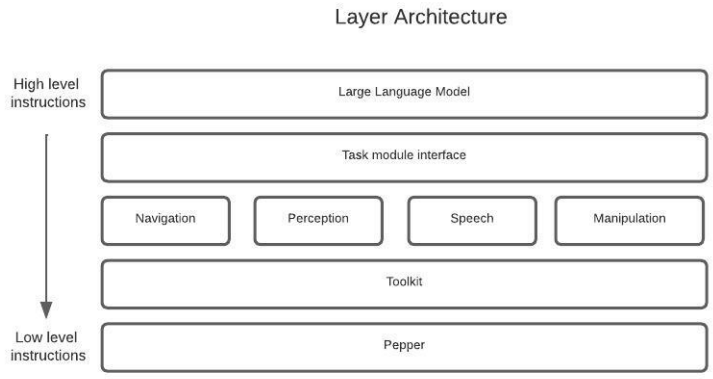


**Fig. 2.** Software Architecture

**Environmental Awareness** Environmental Awareness is an integral part of robotics, especially in dynamic environments such as those found in the RoboCup. It encompasses a robot's ability to perceive and understand its environment, which is necessary for making good decisions and navigating complex environments. In our framework, we include object perception, detection, and mapping to enable the robot to operate autonomously and intelligently within its environment.

On the object perception and detection side, we use YOLO (You Only Look Once) together with CVBridge. YOLO is one of the state-of-the-art deep learning-based methods, which prevails in speed and accuracy in real-time object detection. CVBridge serves as an interface between ROS and OpenCV; it converts images taken by the cameras of the robot into a format easily readable by tools compatible with ROS. Such a setup enables the robot to perceive many objects in its environment, such as obstacles, people, and other robots, thus increasing the number of possible decisions.

Regarding the processes of mapping and localization, the ROS Kinetic Navigation Stack serves as a reliable framework that facilitates path planning and autonomous navigation. The robot builds and maintains an environmental representation using the Navigation Stack, keeps track of its location, and navigates dynamic environments. This integrated approach enables the robot to automatically adapt to changes in the environment, avoid obstacles, and optimize routes in real-time. The combination of YOLO, CVBridge, and the Navigation Stack allows the robot to remain continuously aware of its surroundings, making it a strong contender in RoboCup competitions that require real-time perception and navigation to succeed.

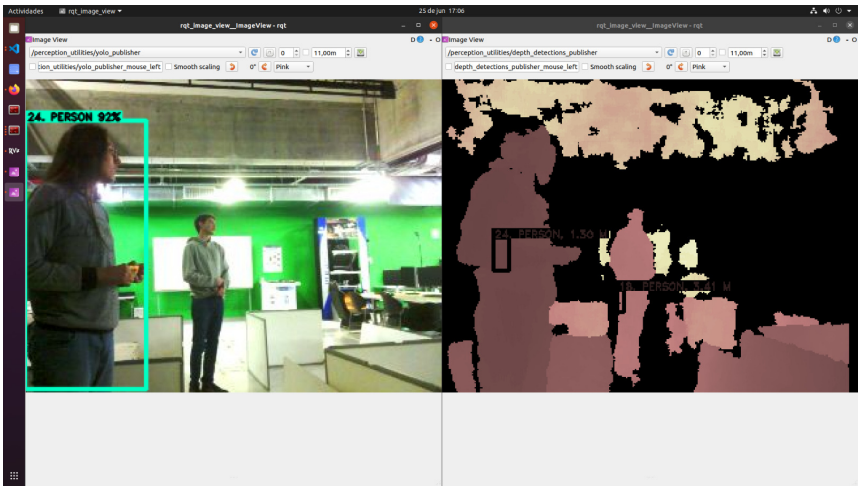


**Fig. 3.** Person detection and distance calculation

**Human-Robot Interaction** Human-Robot Interaction lies at the heart of any autonomous robot, and in this case, social robotics, where the robot is required to understand and be aligned with human instructions. Our HRI approach incorporates speech recognition, object perception, and memory through face recognition in a single framework where the robot can interact with the environment seamlessly. For this, we implemented a system based on Whisper for speech-to-text conversion, complemented with Silero Voice Activity Detection (VAD) for real-time audio evaluation. This backend calls the robot's listening mode at critical moments in time, recording and processing just the speech of interest. Silero VAD makes sure that background noise is excluded, allowing Whisper to recognize the words said by the user and transcribe them into text. This approach improves the robot's capability to hold meaningful conversations while minimizing errors due to background noises. In the case of speech understanding, we followed a hybrid approach that relies on the complexity of the tasks. For simple commands, the robot exploits a set of pre-defined functions that match a limited set of instructions.

On face recognition, the robot employs a technique in which several images of the individual are taken; the system generally takes three distinct photos for identification. The photos are then averaged by computing the median of the 128 dimensional encoding vectors for a proper representation of the individual's face. This is advantageous over averaging the vectors as the median has less sensitivity to outliers. It achieves this by using five images to save a face, and only three are used in the recognition. Stored encodings are then compared against new face images for accurate identification of individuals, such that the robot can recognize and remember its users over time.

When more complex tasks are considered, such as those required in the General Purpose Service Robot Task (GPSR), the robot makes use of Large Language Models (LLMs) to understand a wider class of instructions. In line with the contribution of Becerra et al. (2024), which focuses on optimizing task execution in social robots using large language models (LLMs), our system includes advanced natural language processing mechanisms to improve the performance of tasks executed in dynamic and unpredictable environments. The GPSR tasks often involve human-level complex instructions that can noticeably differ depending on contextual dependencies, and our approach addresses the widespread gap between provided instructions and the robot's ability to execute them adequately. With LLMs, we can provide the robot with a better understanding of open-ended commands, improving its autonomy and the quality of responses it can generate. In particular, the study showed that 55% of tasks generated passed the automatic runtime execution assessment, highlighting the significant progress made in improving task execution in robots. The best success rate was achieved by GPT-4, which underlines the effectiveness of using an LLM in complex and context-sensitive tasks. This double-focal approach allows the robot to handle both structured, systematic tasks and more flexible, conversational interactions, significantly enhancing its practical applicability in real-world settings.

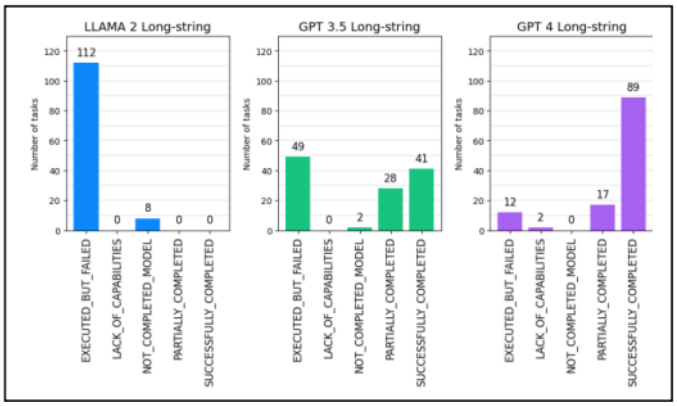


**Fig. 4.** Task results [4]. Reproduced from [4], with permission.

In our system for operator following, the robot detects, tracks, and follows a single operator through a combination of visual detection and motion control. The algorithm relies on data from Pepper's camera to locate the closest person in its visible range. By estimating the position and size of the person's bounding box, the robot calculates the distance and angle to the operator. The robot then adjusts its movement—both linear and angular—ensuring it maintains a safe following distance while reacting to changes in proximity.

The robot's motion is controlled based on these calculations:

**Inputs:**

- $max\_width$: Maximum width of the detected person (typically 140 pixels)
- $person\_width$: Current width of the detected person
- $speed$: Predefined speed factor
- $followed\_person[1]$: X-coordinate of the left edge of the person
- $current\_head\_angle$: The current yaw angle of the robot's head

**Outputs:**

- $calculated\_vel$: Robot's linear velocity
- $angular\_vel$: Robot's angular velocity

**Algorithm:**

1. **Linear Velocity Calculation:**
   Calculates the robot's movement speed based on the width of the detected person, ensuring a minimum speed even when the person is close.

$$calculated\_vel = \begin{cases} speed \times (max\_width - person\_width) + 0.05 & \text{if } person\_width < max\_width \\ 0.01 & \text{otherwise} \end{cases}$$

2. **Person's Center Calculation:**
   Computes the center of the detected person based on their left edge and width to determine their position in the camera frame.

$$person\_center = followed\_person[1] + \frac{person\_width}{2}$$

3. **Yaw Angle Calculation:**
   Converts the person's position (in pixels) into an angular value (in degrees) relative to the robot's field of view, considering the robot's head angle.

$$person\_degree\_yolo = (person\_center \times 0.16875) - 27$$

4. **Angle Difference:**
   Determines the difference between the detected person's angle and the robot's current head angle to align the robot's orientation.

$$person\_degree = person\_degree\_yolo - current\_head\_angle$$

5. **Angular Velocity Calculation:**
   Determines the rotational speed needed to align the robot with the detected person's yaw angle, taking into account a scaling factor to limit the robot's turning speed.

$$angular\_vel = -\min\left(0.5, \frac{person\_degree}{|person\_degree|} \times 0.5\right)$$

**Explanation of Constants:**

- $max\_width = 140$ pixels: Chosen based on expected person size at average distance.
- $0.05m/s$: Ensures a minimum speed even when the person is very close.
- 0.16875: Scaling factor to convert pixels to degrees based on camera field of view.
- $-27°$: Offset in degrees to align the detected person's position with the robot's frame.
- $0.5m/s$: Controls the robot's rotation speedin meters per second.

The system also includes obstacle avoidance algorithms and recalibration methods, enabling the robot to reroute or reverse in response to obstacles, while speech functionalities keep the operator informed about the robot's status or requests for help.

Combining speech recognition, natural language processing, and perceptual awareness, these integrated systems give the robot the ability to understand and respond to human interaction in a more appropriate manner, thereby creating a more natural and intuitive human-robot interaction experience.

**World Interaction** World interaction is a critical aspect of social robotics, especially in dynamic environments where the robot must navigate and manipulate objects.

The quality of the camera and the manipulation capabilities of Pepper's arms are quite poor and limited, which seriously impairs the capabilities when performing manipulation tasks. The limited range of the depth camera, together with the ineffective grasping capabilities of the robot, makes the implementation of conventional manipulation strategies—such as trajectory planning based on accurate targeting of the end-effector—difficult. As a result, we changed our plan to rely on human operators' assistance using the "Deus Ex Machina" clause allowed by RoboCup rules. This allowed us to request an operator to place an object directly directly onto the robot's open hands. In addition, we developed specific poses that allowed Pepper to securely grasp objects once they had been placed accurately on its hands thus saving the need for complex manipulation planning. While this was not ideal, it proved to be the most effective way of working within the restrictions of the robot in the competition.

Finally, the ROS Navigation Stack was run directly on Pepper's processor to minimize latency. Our navigation setup uses AMCL, move base, global planner, and many other essential elements from the ROS Navigation Stack to enable reliable navigation in dynamic environments.

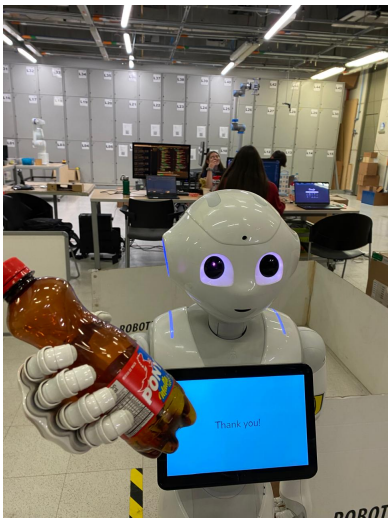


**Fig. 5.** Succesful grasp

## 5 Current Research

**Enhancing Autonomous Task Execution through Large Language Models** Our work follows that of Becerra, Colmenares, and Manrique (2024) in improving autonomous tasks of social robots using Large Language Models. In the present research, deep integration of LLMs is performed to advance autonomy and task execution by social robots, like Pepper in complex dynamic scenarios. The key research focus is towards developing the robot's understanding, planning, and execution of tasks carried out autonomously. That builds up to expand the abilities beyond rigid, predefined task structures to more flexible, context-aware task management.

We believe that, by integrating LLMs, we can enhance the instruction-interpreting capability of the robot, making it capable of performing tasks with greater flexibility in response to changing environmental clues.

The robot achieves this by developing a strong context-awareness system that enables it to understand situational context, predict what should come next in a task, and adapt its behavior accordingly. This is made possible through a systematic methodology that involves three core steps. First, verbal instructions are transcribed into text, enabling the robot to analyze and process them. Second, the transcribed text is processed by a large language model, which generates precise code instructions tailored for the robot's execution. Finally, the robot carries out the task by executing these instructions.

This approach therefore leverages the power of computation that LLMs bring about to enhance the decision-making processes of the robot so that at its maximum capacity, the robot can do not only instruction following but also real-time adaptation on new tasks. Additionally, the research establishes how LLMs can be trained to identify and understand ambiguous or incomplete instruction for enhanced handling of dynamic and unpredictable situations by a robot.

**Future Research: Towards Continuous Autonomy with AutoGPT** Following our victory in the RoboCup@Home Social Standard Platform League this year, a new generation of work will center on combining AutoGPT, an open-source AI agent that autonomously breaks down goals into sub-tasks and executes them in a continuous loop, in an effort to harness the most significant level of autonomy. This approach allows the robot to set its own agenda and keep learning to respond to new and evolving environments, rather than relying on a user to provide a defined task. AutoGPT presents a possibility: give the robot the ability of self-improvement of its abilities over time, learning from past interactions and adapting to new challenges on its own, without human intervention.

The present research will attempt to give the robot the capability to become an independent agent that can recognize new problems and solve them on its own in real time. With AutoGPT, it will be possible for the robot to judge which tasks have to be done independently, manage its priorities, and find novel solutions in complex and uncertain environments. When the reasoning and decision-making

abilities of LLMs are combined with the generative powers of AutoGPT, the robot will be capable of working independently with minimal supervision in a variety of settings. This may redefine the role that social robots are going to play and will be more capable of withstanding long-term engagements in dynamic environments where they can take responsibilities autonomously, adapt to shifting contexts, and provide consistent service to users. The role of robots has evolved from being merely task executors to intelligent, autonomous agents that understand, learn, and respond to their environment in real time, calibrating their behavior to better serve needs and changes in the environment.

## 6 Summary and Outlook

SinfonIA Uniandes is pioneering work in social robotics, focused on improving the interaction capabilities of robots like Pepper. Our current research combines state-of-the-art technologies in perception, speech recognition, and autonomy to enable robots capable of engaging in flexible, open-ended interactions with human users. The objective of this research is to enhance the capacity of robots to comprehend and react to their surroundings, identify individuals, retain memories of prior interactions, and adjust to various social contexts. Such improvements are crucial for rendering robots more human-like, relatable, and proficient within social settings.

By using cloud-based multimodal models and distributed systems like ROS, we enable efficient robot operation, even with limited on-board computing resources. The method is crucial in bringing up the accessibility and performance of social robots in highly variable global settings, particularly in developing nations, where the existing infrastructure may not be able to support the advanced robotics computational requirements.

Looking ahead, SinfonIA Uniandes will be devoted to taking the functionalities of social robots further in terms of autonomy and personalized interaction. We envision a future where robots can make decisions independently while being adapted through continuous interaction, engaged in meaningful and contextually relevant conversations. While we work on expanding the frontiers of social robots, our aim is to contribute to the global conversation on the future of interaction among humans and robots, while ensuring that such technologies become usable and beneficial to people everywhere.

**Acknowledgments**: We would like to express our deepest gratitude to the Universidad de los Andes for their unrelenting support, to all members of RoboCup for giving us this great opportunity, and to Luccas Rojas and Juan Jose Garcia for their invaluable guidance throughout the process.

## References

1. SoftBank Robotics, "Pepper Robot," [Online]. Available: https://corporate-internal-prod.aldebaran.com/themes/custom/softbank/images/full-pepper.png.webp. [Accessed: Nov. 17, 2024].

2. OpenAI, "Whisper: A general-purpose speech recognition model," GitHub repository, 2024, available at: https://github.com/openai/whisper.
3. Silero Team, "Silero VAD: pre-trained enterprise-grade Voice Activity Detector (VAD), Number Detector, and Language Classifier," GitHub repository, 2024, available at: https://github.com/snakers4/silero-vad, email: hello@silero.ai.
4. L. R. Becerra, J. R. Colmenares and R. Manrique, "Enhancing Autonomous Task Execution in Social Robots with Large Language Models," 2024 10th International Conference on Automation, Robotics and Applications (ICARA), Athens, Greece, 2024, pp. 40-44, doi: 10.1109/ICARA60736.2024.10552978.
5. United Robotics Group, "Pepper Technical Specifications," United Robotics Group Support, [Online]. Available: https://support.unitedrobotics.group/en/support/solutions/articles/80000958735-pepper-technical-specifications. [Accessed: Dec. 6, 2024].
6. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," arXiv preprint arXiv:1506.02640, Jun. 2015. [Online]. Available: https://arxiv.org/abs/1506.02640. [Accessed: Dec. 6, 2024].
7. R. Memmesheimer, V. Seib, and D. Paulus, "homer@UniKoblenz: Winning Team of the RoboCup@Home Open Platform League 2017," in RoboCup 2017: Robot World Cup XXI, H. Akiyama, O. Obst, C. Sammut, and F. Tonidandel, Eds., vol. 11175, Lecture Notes in Computer Science. Cham: Springer, 2018, pp. 514-524, doi: 10.1007/978-3-030-00308-1_42.
8. D. Holz, J. Ruiz del Solar, K. Sugiura, and S. Wachsmuth, "On RoboCup@Home – Past, Present and Future of a Scientific Competition for Service Robots," Academia.edu, [Online]. Available: https://www.academia.edu/93013980/On_RoboCup_at_Home_Past_Present_and_Future_of_a_Scientific_Competition_for_Service_Robots. [Accessed: Dec. 6, 2024].
9. J. Hart, A. Moriarty, K. Pasternak, J. Kummert, A. Hawkin, V. Hassouna, J. D. Pena Narvaez, L. Ruegemer, L. von Seelstrang, P. Van Dooren, J. J. Garcia, A. Mitzutani, Y. Jiang, T. Matsushima, and R. Polvara, "RoboCup@Home 2024: Rules and Regulations," GitHub, [Online]. Available: https://github.com/RoboCupAtHome/RuleBook/releases/tag/2024.1. [Accessed: Dec. 6, 2024].
10. M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating System," in ICRA Workshop on Open Source Software, 2009.
11. D. Holz, J. R. del Solar, K. Sugiura, and S. Wachsmuth, "On RoboCup@Home – Past, Present and Future of a Scientific Competition for Service Robots," in Lecture Notes in Computer Science, vol. 9348, Springer, 2015, pp. 686-697, doi: 10.1007/978-3-319-18615-3_56.
12. M. Matamoros, V. Seib, and D. Paulus, "Trends, Challenges and Adopted Strategies in RoboCup@Home," CoRR, vol. abs/1903.02516, 2019. [Online]. Available: http://arxiv.org/abs/1903.02516. [Accessed: Dec. 6, 2024].
13. Significant Gravitas, AutoGPT: A collection of tools and experimental open-source attempts to make GPT-4 fully autonomous. [Online]. Available: https://github.com/Significant-Gravitas/AutoGPT. [Accessed: Dec. 6, 2024].