



**SAPIENZA**  
UNIVERSITÀ DI ROMA

## Enhancing Human–Robot Interaction with Personality and Individual Preferences

Department of Computer, Control and Management Engineering, Sapienza University of Rome  
Master's Degree Program in Artificial Intelligence and Robotics

**Alessia Tancredi**  
ID number 2076114

Advisor  
Prof. Luca Iocchi

Co-Advisor  
Alessio Saladino

Academic Year 2024/2025

---

**Enhancing Human–Robot Interaction with Personality and Individual Preferences**  
Sapienza University of Rome

© 2025 Alessia Tancredi. All rights reserved

This thesis has been typeset by L<sup>A</sup>T<sub>E</sub>X and the Sapthesis class.

Author's email: tancredi.2076114@studenti.uniroma1.it

# Abstract

This thesis presents a human–robot verbal interaction framework that combines natural language processing with movement control, facial expressions, and real-time visual perception, enabling a social robot to communicate fluently, respond accurately to environmental questions, and adapt its physical behavior to the context. While large language models (LLMs) integrated into robots excel at providing factual information and executing contextually appropriate actions, they typically avoid expressing opinions, emotions, or preferences, resulting in neutral and impersonal interactions.

To address this limitation, this work introduces a personality and preference model that allows each robot to embody a distinct character, such as being more reserved or more extroverted, and to maintain consistent likes, dislikes, and conversational styles. These traits influence both verbal and non-verbal communication, ensuring that the robot’s behavior reflects its personality in a coherent and believable manner. By integrating this model into the multimodal interaction framework, the robot becomes a more engaging, relatable, and socially present partner, capable of fostering deeper user connection and more natural long-term interaction.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Objectives . . . . .	1
1.3	Chapter Presentation . . . . .	2
<b>2</b>	<b>Background</b>	<b>4</b>
2.1	Social Robots and Human–Robot Interaction . . . . .	4
2.2	Applications of Social Robots . . . . .	4
2.2.1	Education . . . . .	5
2.2.2	Care and Assistance . . . . .	5
2.2.3	Medicine . . . . .	5
2.2.4	Other Emerging Domains . . . . .	6
2.3	Personality and Preferences in Human–Robot Interaction . . . . .	6
2.3.1	Personality . . . . .	6
2.3.2	Preferences . . . . .	7
2.4	Large Language Models in Human–Robot Interaction . . . . .	7
2.4.1	Enhanced Language Understanding . . . . .	7
2.4.2	Multimodal Integration . . . . .	7
2.4.3	Challenges and Open Issues . . . . .	7
<b>3</b>	<b>Tools and Technologies</b>	<b>9</b>
3.1	ROS . . . . .	9
3.2	ChromaDB . . . . .	10
3.3	Docker: Image, Container, and Virtualization . . . . .	11
3.4	GPT-4o Mini . . . . .	12
3.5	MARRtina Robot . . . . .	13
3.5.1	Physical Architecture and Mobility . . . . .	13
3.5.2	Visual and Emotional Expression . . . . .	14
3.5.3	Sensing and Perception . . . . .	14
3.5.4	Software and Control Infrastructure . . . . .	14
3.5.5	Use in Research and Education . . . . .	14
3.6	SMARRtino Robot . . . . .	15
3.6.1	Physical Architecture and Mobility . . . . .	15
3.6.2	Emotional Expression . . . . .	16
3.7	TIAGo Robot . . . . .	16
3.7.1	Physical Architecture and Mobility . . . . .	16
3.7.2	Perception and Sensing . . . . .	17
3.7.3	Software and Integration with AI Components . . . . .	17
3.7.4	Expressivity and Movement . . . . .	17

3.7.5	Role in Research and Deployment . . . . .	17
3.8	NAO Robot . . . . .	18
3.8.1	Physical Design and Degrees of Freedom . . . . .	18
3.8.2	Sensing and Perception . . . . .	18
3.8.3	Software, Control and Middleware . . . . .	19
3.8.4	Expressive Capabilities . . . . .	19
3.8.5	Use in Human–Robot Interaction Studies . . . . .	19
<b>4</b>	<b>Existing Framework Overview</b>	<b>20</b>
4.1	Original Framework . . . . .	20
4.2	Motivation and scope . . . . .	21
4.3	A-Priori Supervisor . . . . .	21
4.3.1	Social Supervisor . . . . .	21
4.3.2	Functional Supervisor . . . . .	22
4.4	Long-term Memory and QA . . . . .	22
4.5	Conversational Assembly . . . . .	23
4.6	Movement Module . . . . .	23
4.7	Emotion Module . . . . .	24
4.7.1	Integration Strategy . . . . .	24
4.7.2	Operational Flow . . . . .	25
4.7.3	Effect on Interaction Quality . . . . .	25
4.8	Breathing Movement Loop . . . . .	25
4.9	Vision module . . . . .	25
4.10	Existing Limitation with Personality and Preferences . . . . .	26
4.10.1	Observed Limitations in Interaction Quality . . . . .	26
4.10.2	Motivation for Introducing Personality and Preference Modelling . . . . .	26
<b>5</b>	<b>Personality and Preferences: Architectural Extension</b>	<b>28</b>
5.1	Overview of the Extended Framework . . . . .	28
5.2	Personality Management and Initialization . . . . .	29
5.2.1	Emotion Policy . . . . .	29
5.2.2	Personality Instruction Block . . . . .	30
5.3	Opinion Classification and Routing . . . . .	31
5.4	Preference-Aware Answerer and Memory Integration . . . . .	32
5.4.1	Confidence Score Computation . . . . .	32
5.4.2	Response Generation and Preference Updating . . . . .	33
5.4.3	Illustrative Example . . . . .	33
5.5	Reasoning Mode and Autonomous Reflection . . . . .	34
<b>6</b>	<b>Architecture Implementation</b>	<b>36</b>
6.1	Personality Traits Retrieval and Prompt Integration . . . . .	36
6.1.1	Parsing Traits at Startup . . . . .	36
6.1.2	From Traits to Emotion Policy . . . . .	37
6.1.3	Building the Personality Instruction Block . . . . .	37
6.1.4	Initiative Based on Verbosity . . . . .	40
6.2	Preference Storage, Deduction, and Updating Mechanisms . . . . .	41
6.2.1	Memory Format and Extraction . . . . .	41
6.2.2	Reasoning Mode: Deducing New Preferences . . . . .	41
6.2.3	Opinion Classification Workflow . . . . .	42
6.2.4	Answering Opinion Questions Consistently . . . . .	43

6.2.5	Confidence Scores . . . . .	44
6.2.6	Writing Back New Preferences . . . . .	46
6.3	Integration with the Existing Vision Module . . . . .	47
6.3.1	Dual Input: Language and Visual Context . . . . .	47
6.3.2	Integrating Visual Context with Dialogue Reasoning . . . . .	47
6.3.3	Implementation Details . . . . .	48
6.3.4	Simulation and Real-Robot Deployment . . . . .	48
6.3.5	Illustrative Scenarios . . . . .	49
6.3.6	Summary . . . . .	49
<b>7</b>	<b>Deployment on Robots</b>	<b>50</b>
7.1	Administration Server – Movement, Emotion, and Expression Control . . . . .	50
7.2	Robot’s Docker Connection . . . . .	51
7.3	Movement Server . . . . .	51
7.4	Emotion Server . . . . .	51
7.5	Robot Movement Management . . . . .	52
7.6	Breathing Movements . . . . .	52
7.6.1	Start and Stop Loop Management . . . . .	52
7.7	Parallel Management of Movement and Voice . . . . .	52
7.8	Facial Expressions . . . . .	52
7.9	“Breathing” Expressions . . . . .	53
<b>8</b>	<b>Realistic Use Cases and Behaviour Analysis</b>	<b>54</b>
8.1	Engaging in Personality-Consistent Dialogue . . . . .	54
8.2	Handling Opinion-Based Questions Effectively . . . . .	56
8.2.1	Example Set 1: Questions Related to Stored Preferences . . . . .	56
8.2.2	Example Set 2: Questions Related to Deductions . . . . .	57
8.2.3	Example Set 3: Questions Without Preferences or Deductions . . . . .	58
8.3	Opinion-Based Interaction with Visual Context . . . . .	60
8.4	Reasoning Mode: Autonomous Preference Inference . . . . .	64
8.4.1	Execution Model . . . . .	64
8.4.2	Memory Format And Update Mechanism . . . . .	64
8.4.3	Cold-start case: no explicit preferences, no deductions . . . . .	68
8.4.4	Methodological Overview and Guarantees . . . . .	69
8.5	Social Presence and User Engagement Improvement . . . . .	69
<b>9</b>	<b>Quantitative Evaluation and Performance Analysis</b>	<b>71</b>
9.1	Opinion Classifier Evaluation . . . . .	71
9.1.1	Dataset Description . . . . .	71
9.1.2	Global Performance . . . . .	72
9.1.3	Class-Specific Evaluation . . . . .	73
9.1.4	Confusion Matrix and Error Distribution . . . . .	74
9.1.5	Qualitative Error Analysis . . . . .	75
9.1.6	Discussion and Interpretation . . . . .	76
9.2	Evaluation of Preference Acquisition and Reasoning . . . . .	76
9.2.1	Preference Acquisition Rate (PAR) . . . . .	76
9.2.2	Deduction Generation Rate (DGR) . . . . .	79
9.2.3	Correlation between Preference Acquisition and Deduction Generation . . . . .	81

*CONTENTS*

v

<b>10 Summary and Conclusion</b>	<b>82</b>
10.1 Summary . . . . .	82
10.2 Future Works . . . . .	83
10.3 Conclusion . . . . .	84
<b>Bibliography</b>	<b>85</b>

# Chapter 1

## Introduction

### 1.1 Motivation

The development of this thesis is primarily motivated by the desire to integrate and evaluate the use of a personality and preference modelling module within an advanced language-based Human–Robot Interaction (HRI) framework. Recent advances in Large Language Models (LLMs) have shown that their integration in HRI can enable natural, fluid, and contextually adaptive verbal exchanges between humans and robots. These models are capable of generating coherent and context-aware responses, allowing interaction to move beyond fixed, keyword-based dialogue toward more natural and humanlike conversation.

However, while LLMs perform remarkably well for factual and procedural queries, they are inherently designed to avoid expressing personal opinions, preferences, or emotions. This design choice, intended to ensure neutrality and safety, often results in interactions that are emotionally flat and lacking in individuality. This limitation became evident during an outreach event in which middle school students were invited to freely converse with the robot. A considerable share of the questions addressed to the robot went beyond factual information, seeking to explore its tastes, opinions or emotions. When it answered by stating that, as a robot, it lacked such qualities, the dialogue became repetitive and unengaging, limiting the possibility of building rapport and sustaining interaction.

This thesis addresses the limitation by assigning each robot a distinct set of personality traits and preferences, including degrees of sociability and specific likes or dislikes. These traits shape both verbal and nonverbal behaviour, allowing the robot’s style of communication, movements and facial expressions to remain consistent with its designated personality throughout interactions. By integrating the personality module into the broader multimodal framework, which already includes movement control, facial expression management and visual perception, the robot shifts from being a neutral conversational partner to a social agent capable of engaging users in a more personal and consistent way.

### 1.2 Objectives

This thesis is guided by objectives that seek to expand the role of social robots in Human–Robot Interaction. The research moves beyond generic dialogue and explores how robots can sustain interaction through a stable and recognisable identity. For this purpose, a framework for personality and preference modelling is proposed, shaping communication style, behaviour and expressive output in a consistent way.

The aim is to allow robots to take part in conversations that are not limited to factual responses

but also convey attitudes and preferences linked to their assigned profile. Such an approach is intended to support longer and more engaging interactions, strengthen user trust, and open opportunities in domains where continuity and emotional connection are central, such as education, entertainment and companionship. The objectives pursued can be summarised as follows:

- Assign each robot a distinct and well-defined set of personality traits, including tone, verbosity, expressiveness, humor, energy level, curiosity, formality, sociality, and patience.
- Enable a dedicated chat mode in which the robot can participate in open-ended conversational exchanges, dynamically adapting to the robot's personality profile.
- Introduce a classification system to determine whether user input expresses or requests a personal opinion.
- Develop mechanisms for the robot to respond to opinion-based questions decisively and consistently with its predefined personality and stored preferences.
- Allow the robot to progressively enrich its personality by incorporating new preferences derived from interactions.
- Provide a reasoning mode in which the robot can autonomously reflect on its stored preferences and deduce new ones.
- Ensure that the robot's personality influences both verbal and non-verbal behaviour, including facial expressions.
- Integrate the pre-existing vision model with the preference module, enabling the robot to connect perception with subjective responses. For instance, the robot can analyse its surroundings through object detection and then use the preference model to express whether it likes or dislikes the identified object.

### 1.3 Chapter Presentation

This thesis is organized into nine chapters, each addressing a specific aspect of the research.

- **Chapter 2 – Background** reviews relevant literature on social robots, their applications and role in HRI. It explores studies on personality and preference modelling, the use of Large Language Models (LLMs) in robotic interaction, and related approaches for integrating personality-aware behaviours into conversational agents.
- **Chapter 3 – Tools and Technologies** presents the hardware and software infrastructure used in the entire framework. It describes ROS as the middleware for coordinating robot components, ChromaDB as a long-term memory system, Docker for containerised deployment, GPT-4o Mini as the lightweight reasoning engine, and the MARRtina robot as the experimental platform. The chapter also discusses the integration of object detection models used to connect the vision module with the preference system.
- **Chapter 4 – Existing Framework Overview** summarises the initial system, describing the conversational pipeline, the functional and social supervisors, and the integration of movement and emotion modules. It also examines the breathing loop mechanism and the vision module developed previously, identifying the major limitation of the framework when operating without personality and preference management.

- **Chapter 5 – Personality and Preferences: Architectural Extension** introduces the proposed architecture. It defines the set of personality traits, explains the design of interaction modes, describes the opinion classification process, and details how the robot generates responses that remain consistent with its assigned personality. Mechanisms for personality growth and preference adaptation over time are also introduced, as well as the integration between the vision system and the preference module, allowing the robot to express opinions about perceived objects.
- **Chapter 6 – Architecture Implementation** provides a detailed technical account of how the framework was realised. It explains personality retrieval and preference storage, the classification pipeline for opinion-based queries, the confidence scoring strategy, and the integration of the personality module with multimodal supervisors handling vision, movement and emotion.
- **Chapter 7 – Deployment on Robots** discusses the implementation of the framework on the physical robot. It details the administration server for managing robot control, the use of the onboard camera for vision-based preference queries, and the integration of supervisors for coordinating verbal, visual and behavioural output during real-world testing.
- **Chapter 8 – Realistic Use Cases and Behaviour Analysis** presents an evaluation of the system in practice. Scenarios include open-ended opinion dialogues, consistency in preference expression, adaptation of communicative style to different users, integration of visual perception with preference reasoning, and the resulting impact on engagement and trust.
- **Chapter 9 – Quantitative Evaluation and Performance Analysis** presents the quantitative assessment of the framework, introducing numerical metrics to evaluate its performance. It analyzes the accuracy of the opinion classifier and measures how different robot personalities influence preference acquisition and reasoning abilities.
- **Chapter 10 – Summary and Conclusion** reviews the objectives of the thesis and how they were achieved. It summarises the framework's outcomes, with a focus on advances in personality and preference modelling for Human–Robot Interaction. The chapter reflects on the broader implications for social robotics and concludes by outlining future work.

# Chapter 2

## Background

### 2.1 Social Robots and Human–Robot Interaction

Social robots are embodied systems created to interact with people through natural and meaningful communication. Their role is not limited to the execution of tasks, but extends to participation in daily activities where collaboration, assistance, and companionship are central. To achieve this, they combine verbal and non-verbal behaviours, interpret human actions and intentions, and adapt to social norms and contextual cues [8, 10].

Human–Robot Interaction (HRI) is the field that studies the models, algorithms, and methodologies that enable this type of interaction. It is inherently multidisciplinary, involving robotics, artificial intelligence, psychology, neuroscience, and design. Research in this area focuses on developing systems able to recognise and respond appropriately to human behaviour, sustain engagement over time, and build trust in different application domains. This requires rethinking classical robotics problems such as perception, planning, and learning in order to address social challenges including turn-taking, intention recognition, and adaptive communication [21, 25]. The physical embodiment of robots strongly influences how they are perceived. While realistic designs may increase familiarity, excessive resemblance to humans without fully matching their appearance and motion can create discomfort, a phenomenon often referred to as the uncanny valley [15]. For this reason, alternative morphologies and stylised forms are frequently adopted to balance acceptance and usability. Moreover, studies show that the physical presence of robots compared to virtual agents can enhance trust, engagement, and learning outcomes, underlining the importance of embodiment in HRI [25].

Finally, the deployment of social robots raises methodological and ethical challenges. Prototyping often relies on partial human control, which complicates reproducibility and transparency if not properly documented [21]. At the same time, their introduction in sensitive contexts such as education and healthcare requires careful consideration of issues related to privacy, safety, long-term dependency, and the social consequences of sustained interaction [27].

In conclusion, social robots and HRI represent the intersection of technical innovation and social understanding. Their progress depends not only on advances in autonomy and learning, but also on the ability to integrate ethical principles, user expectations, and social dynamics into the design of systems that can function effectively in real-world environments.

### 2.2 Applications of Social Robots

The potential of social robots extends far beyond experimental demonstrations, as their ability to sustain natural interaction has been increasingly explored across multiple domains. Their capacity to combine verbal and non-verbal communication, monitor user behaviour, and adapt

their responses makes them suitable for contexts where engagement, trust, and personalization are essential. The following subsections present the main application areas where social robots have been most actively deployed: education, care and assistance, medicine, and other emerging sectors.

### 2.2.1 Education

One of the most widely studied fields for social robots is education. Robots are introduced as tutors, peers, or companions that can deliver learning material in a way that is interactive and engaging. Research shows that their presence often increases motivation, attention, and enjoyment among learners, especially in younger populations. Unlike traditional computer-based learning tools, robots benefit from physical embodiment, which enhances social presence and creates stronger engagement with students [5].

In classroom environments, social robots can guide exercises, repeat explanations, and provide personalised feedback based on the learner's performance. By sensing levels of attention and emotional states, they can dynamically adapt their behaviour to maintain interest and reduce disengagement [2]. This adaptability is particularly important in language learning, mathematics, and collaborative tasks, where sustaining motivation plays a crucial role in the learning process [12]. Moreover, robots can be integrated into remote learning platforms, providing continuity and a sense of presence even when direct interaction with teachers is limited [6]. These findings suggest that robots can complement, rather than replace, human educators by offering personalised support, scaffolding difficult concepts, and enriching the overall learning experience.

### 2.2.2 Care and Assistance

Social robots are also widely adopted in the domains of care and assistance, where they act as companions and helpers for elderly individuals, people with disabilities, and children with special needs. Their ability to combine emotional support with practical assistance makes them valuable tools for promoting independence and improving quality of life. Robots can engage in conversations, provide reminders for daily activities, encourage healthy routines, and act as mediators between users and caregivers [7, 18].

In interventions with children with developmental or intellectual disabilities, robots have proven effective in supporting social and cognitive skills through structured play, exercises, and therapeutic programs [23]. They offer predictable and consistent behaviour, which is often easier for children with autism spectrum disorders to interpret compared to human interaction. For elderly users, social robots are designed to reduce loneliness, stimulate cognitive activity, and encourage social participation, thereby alleviating caregiver burden [30, 4]. These applications highlight the dual role of robots in both emotional and functional support, enabling vulnerable populations to benefit from a more personalised and continuous form of assistance.

### 2.2.3 Medicine

In medical and clinical contexts, social robots are employed as supportive tools for patients and healthcare professionals. They can provide comfort and distraction for children undergoing stressful procedures, reduce anxiety in hospital environments, and assist with adherence to treatments by offering encouragement and guidance [30]. In rehabilitation, robots are used to facilitate repetitive motor training, monitor progress, and deliver feedback to motivate patients throughout their recovery [22].

Social robots have also been tested in psychological interventions, where their non-judgmental nature helps patients engage more openly in therapy sessions. For example, they can participate in cognitive-behavioural exercises, promote adherence to rehabilitation protocols, or act as

mediators in group therapy. Furthermore, robots that provide constructive performance feedback have been shown to positively influence motivation, self-esteem, and emotional states, underlining their potential in therapeutic settings [14]. While current evidence is promising, larger clinical trials and long-term evaluations are required to validate their impact and ensure safe integration into healthcare practice.

#### 2.2.4 Other Emerging Domains

Beyond education, care, and medicine, social robots are increasingly explored in new areas such as tourism, customer service, and sports. In hospitality, they serve as guides and receptionists, providing information, enhancing visitor experiences, and shaping perceptions of innovative service environments [28]. In professional and organizational contexts, robots capable of delivering performance feedback are studied as tools for workforce training, helping employees improve skills and maintain motivation [14]. In sports and fitness, social robots are applied as coaches that can motivate users, encourage adherence to exercise routines, and improve training outcomes by combining feedback with social support [19].

These emerging applications illustrate the versatility of social robots and their ability to generate value in diverse environments. Their capacity to combine functionality with relational engagement suggests that their use will continue to expand into new sectors, driven by the need for more interactive and human-centred technologies.

In summary, the applications of social robots demonstrate their potential to complement human efforts in education, healthcare, assistance, and beyond. Their success depends not only on technical performance but also on their ability to adapt socially, respect ethical boundaries, and provide meaningful support over time in real-world environments.

### 2.3 Personality and Preferences in Human–Robot Interaction

The integration of personality and preference modelling into Human–Robot Interaction (HRI) has emerged as a central research direction for creating robots that are not only functional but also socially competent. These two dimensions contribute to building coherent and engaging behaviours that allow robots to move beyond task execution and become interactive partners capable of sustaining meaningful exchanges with humans. While personality provides the structural foundation for a robot’s style of interaction, preferences allow it to express individuality and attitudes, both of which are critical for fostering long-term engagement and trust [26].

#### 2.3.1 Personality

Personality in robots refers to a set of stable behavioural dispositions that define how interaction unfolds across different contexts. Traits such as extroversion, sociability, patience, or formality shape the way robots communicate, express emotions, and regulate their level of engagement. A robot with a more extroverted personality might use expressive gestures, maintain prolonged eye contact, and adopt a playful conversational style, while a more reserved robot would interact in a concise, calm, and measured way.

Endowing robots with distinct personalities serves several purposes. First, it allows users to more easily predict the robot’s behaviour, which fosters consistency and reliability. Second, it influences how humans perceive the robot, as people naturally attribute social identity and human-like qualities to agents that display recognisable patterns of behaviour. Finally, it opens the possibility for adapting the robot’s interaction style to different user groups, such as children, older adults, or professionals, thus increasing acceptance and usability in a wide range of domains [26].

### 2.3.2 Preferences

While personality provides a general behavioural framework, preferences allow robots to display attitudes, likes, and dislikes in ways that make interaction more personal and engaging. Expressing preferences transforms the robot from a neutral information provider into an entity that can participate in opinion exchanges and social dialogue. For instance, when asked whether it enjoys a certain activity or object, a robot with defined preferences can respond consistently, reinforcing its individuality.

The modelling of preferences is closely tied to user expectations. People interacting with robots often ask questions about their opinions, tastes, or emotions, and the absence of such qualities may result in repetitive or unengaging conversations. By embedding structured preferences, robots can overcome this limitation, sustaining dialogues that feel more natural and relatable. Moreover, preferences can evolve over time through repeated interactions, enabling robots to enrich their character and build a sense of continuity in long-term engagements.

Integrating preferences with other modules, such as vision or affective computing, further enhances the interaction. For example, a robot that perceives an object in its environment and expresses a positive or negative stance toward it combines perception with personality-driven reasoning. This capacity strengthens the illusion of individuality and supports a deeper form of social presence in human–robot encounters [26].

## 2.4 Large Language Models in Human–Robot Interaction

Large Language Models (LLMs) have recently become a cornerstone for advancing Human–Robot Interaction (HRI). Their integration provides robots with the ability to interpret, reason, and communicate in ways that extend far beyond traditional dialogue systems. Instead of relying on rigid rules or keyword matching, LLMs enable more fluid and adaptive exchanges that resemble natural conversation [24].

### 2.4.1 Enhanced Language Understanding

A major contribution of LLMs is their capacity to process the ambiguity of everyday language. Human communication is often incomplete, context-dependent, or vague, making it difficult for conventional systems to interpret. LLMs, trained on large-scale datasets, can infer intent from such inputs and translate them into structured actions. This capability is crucial for collaborative activities where instructions must adapt to dynamic environments and evolving goals [3].

### 2.4.2 Multimodal Integration

Recent work highlights the benefits of combining LLMs with other modalities such as vision, speech, and sensorimotor data. In such architectures, the linguistic reasoning of LLMs is complemented by perceptual inputs, allowing robots to ground language in the physical world. This enables robots not only to interpret commands but also to discuss perceived objects, explain their actions, and adapt their behaviour according to contextual cues. Multimodal integration thus expands the scope of interaction, supporting richer and more flexible exchanges [11].

### 2.4.3 Challenges and Open Issues

Despite these advances, several challenges remain. Current LLM-based systems still face limitations in maintaining long-term memory, consistent identities, and contextually grounded reasoning. Furthermore, ethical concerns arise regarding trust, bias, and accountability in decision-making. Since LLMs can generate convincing but occasionally inaccurate responses, ensuring transparency

and reliability is especially important in sensitive domains such as healthcare, education, and assistance [16].

## Chapter 3

# Tools and Technologies

The implementation of this thesis project relies on a carefully selected set of tools and technologies that enable the integration of a multimodal interaction framework with a personality-aware large language model. These components are responsible for communication, memory, deployment, language reasoning, and robotic embodiment. This chapter details each of these core elements and describes how they contribute to the system's overall functionality and extensibility.

### 3.1 ROS

The Robot Operating System (ROS) is an open-source framework developed to support the creation of robotic applications. Rather than functioning as a traditional operating system, it provides a middleware infrastructure that facilitates communication between software components, hardware abstraction, and system configuration. Its main strength lies in modularity and reusability, allowing developers to design independent components that can be easily integrated into complex robotic systems [20].

ROS is organized around the concept of *nodes*, which are independent processes responsible for specific tasks such as sensor data acquisition, motor control, or decision-making. Nodes communicate through a publish–subscribe mechanism based on *topics*, which serve as named channels for message exchange. This asynchronous and decoupled communication model allows nodes to interact without requiring direct knowledge of each other, thereby improving modularity, scalability, and code reusability. In addition, ROS provides *services* for synchronous request–response communication and a *parameter server* for storing configuration variables at runtime [9].

Another key feature of ROS is its extensive ecosystem of tools and libraries that support the entire development cycle of robotic applications. These include simulation environments such as Gazebo, visualization tools like RViz, and utilities such as rosbag for data recording and analysis. The framework is further enriched by a large set of community-developed packages that implement state-of-the-art algorithms in perception, navigation, manipulation, and control, which can be directly reused or extended. This collaborative environment accelerates development, reduces costs, and promotes standardization across different robotic platforms [29].

The evolution of ROS led to the development of ROS 2, designed to address the limitations of the original system. While ROS 1 was widely adopted in academia, it lacked built-in features for robustness, security, and distributed large-scale deployment. ROS 2 introduces a redesigned communication layer based on the Data Distribution Service (DDS) standard, offering real-time capabilities, configurable quality-of-service (QoS) parameters, lifecycle management of nodes, and support for integration with microcontrollers (micro-ROS). These improvements make it suitable for industrial, safety-critical, and multi-robot applications, bridging the gap between

academic research and commercial deployment [29].

In the existing framework, ROS has been employed as the middleware to coordinate the perception, decision-making, and actuation modules of the robot. Thanks to its flexible and modular design, it enabled seamless integration of the system’s components, ensuring reliable communication, efficient execution of tasks, and a structured approach to managing the robot’s behaviour. Overall, ROS represents a de facto standard in modern robotics, providing a powerful ecosystem that continues to evolve and expand within both academic and industrial contexts.

### Key Features of ROS

- **Open-source middleware:** Provides hardware abstraction, inter-process communication, and configuration management for robotic applications.
- **Node-based architecture:** Independent processes (*nodes*) handle specific tasks such as sensing, actuation, or planning.
- **Publish–subscribe model:** Nodes exchange data asynchronously through *topics*, supporting modularity and scalability.
- **Additional communication:** *Services* for synchronous request–response interaction and a *parameter server* for runtime configuration.
- **Rich ecosystem:** Includes tools (RViz, Gazebo, rosbag) and community-developed packages implementing state-of-the-art algorithms.
- **Collaboration and reusability:** Promotes code sharing and rapid prototyping, accelerating research and development.
- **ROS 2 advancements:** DDS-based communication, real-time support, lifecycle nodes, quality-of-service (QoS), and integration with micro-ROS for industrial use.



**Figure 3.1.** ROS (Robot Operating System)

## 3.2 ChromaDB

ChromaDB is an open-source *vector database* designed to efficiently store and retrieve high-dimensional embeddings. Embeddings are numerical representations of data, such as text, images, or audio, that capture their semantic meaning in a form interpretable by machine learning models. This allows systems to compare inputs based on meaning rather than literal or syntactic similarity, representing a major difference from traditional relational databases that rely on structured fields or keywords.

Unlike SQL-based databases, where queries are executed using explicit attributes, ChromaDB enables similarity-based retrieval by comparing embedding vectors. To perform such queries, data

must first be transformed into embeddings using an appropriate model. ChromaDB provides its own embedding model, *all-MiniLM-L6-v2*, but also supports the integration of more powerful alternatives. Each data item is mapped to a fixed-length vector in a high-dimensional space, and queries are transformed into vectors using the same model. The query vector is then compared with stored embeddings using distance functions such as cosine similarity or Euclidean distance. The system returns the  $N$  closest vectors as the most semantically relevant results, where  $N$  is specified by the query.

Beyond similarity search, ChromaDB supports metadata storage associated with embeddings, enabling refined filtering and more complex queries. Additionally, it allows multimodal representation, making it possible to store and retrieve both textual and visual data in a shared vector space, thereby supporting advanced search scenarios that span across modalities.

In this project, ChromaDB is used to manage the robot’s long-term memory by storing knowledge and contextual information that guide its behaviour. Through similarity-based retrieval, the system can quickly access relevant past interactions or contextual knowledge, which improves the accuracy and coherence of responses while reducing the likelihood of hallucinations by the language model. Thanks to its scalability and real-time performance, ChromaDB is well suited for dynamic environments and is particularly effective in the human–robot interaction scenarios addressed in this work.



**Figure 3.2.** Chroma DB

### 3.3 Docker: Image, Container, and Virtualization

Docker is an open-source platform that simplifies the development, deployment, and execution of applications by encapsulating them within standardized units called *containers*. A container includes all the essential components required to run an application, such as source code, system tools, libraries, and dependencies, ensuring that the application behaves consistently across different computing environments.

Unlike traditional Virtual Machines (VMs), which emulate an entire operating system, Docker containers are lightweight and share the kernel of the host operating system. This design drastically reduces resource consumption and allows containers to start up much faster than VMs. As a result, Docker provides an efficient and scalable solution for running applications in isolated environments, while also ensuring portability across different machines and operating systems.

The process of *containerization* in Docker involves three main components:

- **Dockerfile:** A text file that defines the set of instructions needed to build a Docker image. It specifies the base operating system or image, the application code, the required dependencies, and configuration commands for setup and execution.
- **Docker Image:** A read-only template generated from a Dockerfile. It packages together the operating system layers, libraries, and application code. Images can be stored in local repositories or shared via cloud-based registries for reuse and distribution.
- **Docker Container:** A runnable instance of a Docker image. Each container provides an isolated execution environment for the application. Containers can be started, stopped, and deleted as needed. It is important to note that containers are ephemeral: any changes made inside them during runtime are lost once they are terminated, unless explicitly saved.

The workflow of building and running a Docker container begins with creating a Dockerfile. Using the command `docker build -t image_name .`, the Dockerfile is processed to produce a Docker image. This image can then be executed with the command `docker run image_name`, which launches a container instance in an isolated environment. If the image is not present locally, Docker automatically retrieves it from a public or private repository.

The main advantage of Docker is its ability to guarantee *consistency* across environments. Because all dependencies are encapsulated within the container, developers and operators can be confident that the application will run identically on a local machine, on a server, or in the cloud. Furthermore, Docker's lightweight architecture enables scalable deployments, allowing systems to run multiple containers simultaneously with minimal overhead.

In this project, Docker is employed to package and deploy the robotic framework in isolated containers. This ensures reproducibility of experiments, reduces compatibility issues, and allows seamless integration of different modules of the system, ranging from perception to decision-making and actuation.



**Figure 3.3.** Docker.

## 3.4 GPT-4o Mini

GPT-4o Mini is a lightweight variant of OpenAI's GPT-4o model, designed to provide advanced natural language understanding and generation while optimizing computational efficiency. Its main advantages are the reduced latency, the scalability for real-time applications, and a large context window of up to 128,000 tokens, making it particularly suited for interactive systems such as human–robot interaction frameworks.

The model is accessed via the OpenAI API, which requires authentication through an API key obtained via subscription. Once authenticated, user inputs can be sent as structured messages, and the model generates responses in natural language. This design allows developers to integrate GPT-4o Mini into different modules by defining prompts that specify tasks, expected output formats, and behavioral guidelines.

In this thesis, GPT-4o Mini was integrated as the central reasoning engine across multiple supervisory modules, including dialogue management, database question answering, and functional supervision. Each module interacts with the model through a custom Python class (`Gpt_client`), which manages API calls, message formatting, and local conversation history. Since the OpenAI API does not store dialogue history natively, the class maintains a rolling memory of the most recent  $N$  interactions. This ensures contextual coherence in multi-turn conversations while controlling computational costs, as longer histories increase the number of tokens processed. The first element of the history is always the system prompt, ensuring that the model consistently follows its assigned role.

This implementation provides flexibility, as different supervisors can be customized with specific prompts tailored to their tasks, while still relying on the same underlying model. For example, the conversation module focuses on natural exchanges with users, while the functional supervisor evaluates the appropriateness of actions. By encapsulating this logic, GPT-4o Mini ensures coherent communication, controlled memory handling, and adaptable behavior across different contexts.

Moreover, the integration of GPT-4o Mini supports multimodal interaction. By combining its reasoning capabilities with other modules such as perception and preference modeling, the robot can go beyond factual responses and engage in contextual, opinion-based dialogue. The model's ability to process large contexts also supports consistency in longer interactions, reducing repetitiveness and enhancing user engagement.

Overall, GPT-4o Mini played a fundamental role in this work, serving not only as a conversational tool but also as the core reasoning component enabling the robot to adapt its behavior dynamically and maintain socially engaging interactions in real time.



**Figure 3.4.** OpenAI's GPT 4o mini

## 3.5 MARRtina Robot

MARRtina is a humanoid robotic platform developed as part of the open-source MARRtino Project, an educational and research initiative led by Sapienza University of Rome. Designed to serve as an accessible, customizable, and robust platform for experimentation in Human–Robot Interaction (HRI), MARRtina represents an evolution from simple mobile robots to socially capable, context-aware agents.

### 3.5.1 Physical Architecture and Mobility

MARRtina is built on a mobile base that enables it to autonomously navigate indoor environments. The base combines two motorized wheels for controlled directional movement and two passive

castor wheels for enhanced stability and maneuverability. This configuration allows the robot to move smoothly and safely in structured environments such as laboratories, classrooms, and homes.

The robot's torso includes a pair of articulated arms with four degrees of freedom (DOF) each. Although limited in precision, these arms are sufficient to support expressive gestures such as waving, pointing, and symbolic movements, which are important elements in natural HRI.

The head of MARRtina features two DOFs (pan and tilt), allowing the robot to simulate head movements such as nodding and turning. These motions enhance both camera orientation and the expression of attentiveness and social engagement during conversation.

### 3.5.2 Visual and Emotional Expression

One of MARRtina's most distinctive features is its digital animated face, displayed on a 5:4 LCD screen embedded in the head. This face is capable of rendering a variety of facial expressions, enabling the robot to convey emotions such as happiness, curiosity, confusion, or sadness.

These visual expressions are synchronized with verbal and gestural behavior to create a coherent and believable multimodal interaction experience. For example, a cheerful answer may be accompanied by a smiling expression, while a confused response may be paired with tilted head motion and a puzzled look.

### 3.5.3 Sensing and Perception

MARRtina is equipped with a camera located on its forehead, which serves as its primary vision sensor. The camera allows the robot to perceive the environment, detect objects, recognize humans, and monitor its own state. These capabilities are crucial for tasks such as verifying that it is safe to perform gestures, detecting obstacles, and grounding dialogue in the observed environment.

### 3.5.4 Software and Control Infrastructure

MARRtina runs on the Robot Operating System (ROS), which enables modular, distributed control of its various subsystems. ROS nodes handle perception, motion, emotion display, and dialogue management, while inter-process communication is achieved via topics, services, and actions.

The robot's architecture is tightly integrated with modern AI components, including Large Language Models (LLMs) and YOLOv5 for object detection. This enables MARRtina to participate in fluid, context-aware verbal exchanges, reason about user intent, and act accordingly through movement and emotion expression.

The interaction pipeline includes dedicated supervisors, such as the *vision supervisor*, *functional supervisor*, *social supervisor*, and *awake supervisor*, which collaborate to determine the robot's final output at each step of interaction. These supervisors leverage LLM capabilities to evaluate context and generate safe and appropriate actions.

### 3.5.5 Use in Research and Education

Thanks to its **open-source** design, **ROS compatibility**, and **rich sensory-motor features**, MARRtina has become a valuable platform for both research and teaching in social robotics. It supports experimentation in areas such as:

- Emotionally expressive dialogue and behavior
- Vision-based perception and multimodal grounding

- Safety monitoring and contextual reasoning
- Personalized interaction and preference modeling
- Long-term social engagement strategies

Its modular and extensible design makes it especially well suited for academic projects, where researchers and students can customize behaviors, test new algorithms, and explore ethical considerations in human-centered AI systems.



**Figure 3.5.** MARRtina robot.

## 3.6 SMARRtino Robot

SMARRtino is a compact and portable humanoid robot derived from the MARRtina platform. It was designed within the same open-source MARRtino Project at Sapienza University of Rome, with the goal of providing a lightweight, easily transportable version that preserves the expressive and interactive features of its larger counterpart. Although smaller in scale and mechanically simplified, SMARRtino maintains full integration with the same software architecture, allowing it to participate in multimodal human–robot interaction experiments.

### 3.6.1 Physical Architecture and Mobility

SMARRtino shares the overall structural layout of MARRtina but in a more compact form factor. Its mobile base enables controlled navigation in indoor environments, while its reduced size allows it to operate in confined spaces such as classrooms, offices, or exhibition areas. Each arm is equipped with a single degree of freedom (DOF), allowing limited movement for symbolic gestures such as simple waving or lifting motions. Although its mechanical range is more restricted than MARRtina’s four-DOF arms, these minimal gestures are sufficient to accompany speech and

visually signal engagement during interaction.

The robot's head retains the pan-tilt mechanism, supporting horizontal and vertical motion to orient toward users or objects of interest. This mobility, combined with facial expressiveness, allows SMARRtino to simulate social attention despite its simplified kinematics.

### 3.6.2 Emotional Expression

As with MARRtina, SMARRtino features a digital animated face displayed on an LCD screen embedded in its head. The face can reproduce a variety of emotions such as happiness, curiosity, surprise, or confusion, providing a rich non-verbal communication channel. These facial animations are synchronized with verbal output and basic gestures, ensuring consistency between linguistic and visual cues. This design preserves the affective dimension of interaction, even in the absence of complex body motion.



**Figure 3.6.** SMARRtino robot.

## 3.7 TIAGo Robot

TIAGo (Task-Inspired Agile General-purpose robot) is a mobile manipulation platform developed by PAL [17]. Designed as a versatile, service-oriented robot for research and real-world applications, TIAGo combines mobility, dexterity, perception, and modular architecture to support advanced Human–Robot Interaction (HRI) experiments and integration in domestic, industrial, and assistive contexts.

### 3.7.1 Physical Architecture and Mobility

TIAGo uses a differential drive base, with two motorized wheels for propulsion and steering, and additional caster wheels for balance and stability. This configuration allows TIAGo to navigate indoor spaces with high maneuverability, including corridors, offices, and homes. The

base supports safe and flexible movement in cluttered or semi-structured environments. Above the mobile base, TIAGo includes a torso with a 7-degree-of-freedom (DOF) robotic arm. This manipulator allows the robot to reach, grasp, and manipulate objects in three-dimensional space. The arm is designed for human-scale tasks, such as fetching items, opening doors or cabinets, and performing assistive manipulation in everyday environments.

In addition to the arm, TIAGo has a prismatic (vertical) lift that enables changes in height, thus increasing its effective reach range both upward and downward. This allows the robot to interact with furniture, shelves, or objects at different altitudes without reorientation of the base.

The head of the robot houses sensors and vision systems, mounted on a pan-tilt unit that enables flexible camera positioning and environmental scanning.

### 3.7.2 Perception and Sensing

TIAGo is equipped with various sensors to perceive its surroundings. It typically includes one or more RGB-D (color + depth) cameras, time-of-flight sensors or 3D depth cameras, and lidar or laser scanners for mapping and obstacle detection. These sensory modalities support obstacle avoidance, simultaneous localization and mapping (SLAM), environment mapping, human detection, and object recognition. The robot uses these perception modules to feed its decision-making pipeline and to ground dialogue in the visible world.

### 3.7.3 Software and Integration with AI Components

TIAGo is developed using the Robot Operating System (ROS) infrastructure, enabling modular software architecture, distributed processing, and communication through topics, services, and actions. The software stack includes modules for navigation, manipulation planning, perception, and task execution.

For HRI scenarios, TIAGo can integrate with autonomous reasoning, large language models (LLMs), and object detection algorithms to enable context-aware dialogue, visual grounding, and interactive manipulation. In a personality-aware extension (such as in your framework), TIAGo could host the same supervisors (vision, social, functional, etc.) and the new modules (Personality Manager, Opinion Manager, Reasoning Module), making it a generalized embodiment for experiments in socially intelligent robotics.

### 3.7.4 Expressivity and Movement

Although TIAGo's design emphasizes functional utility, it supports expressive behavior through coordinated movement, gesture control, and head/camera motion. For example, head nods, camera tilts, and manipulator gestures can be synchronized with verbal output to produce socially meaningful cues. The robot's motion planners respect safety, smoothness, and human-awareness constraints, ensuring that expressive acts remain socially comfortable and non-intrusive.

### 3.7.5 Role in Research and Deployment

TIAGo has been adopted in many research institutions and practical deployments. Its flexibility, modular architecture, and strong ROS support make it well suited for projects in manipulation, assistive robotics, service robotics, and human–robot interaction. In the context of your framework, TIAGo provides an excellent alternative embodiment for your personality and preference system, enabling comparative studies of personality consistency across different robotic platforms.



**Figure 3.7.** TIAGo robot (source: PAL Robotics).

## 3.8 NAO Robot

NAO is a humanoid robotic platform developed by SoftBank Robotics (formerly Aldebaran Robotics), widely adopted in research, education, and human–robot interaction projects. The current version, NAO6, builds upon years of evolution to deliver improved performance, robustness, and flexibility for social robotics applications [1].

### 3.8.1 Physical Design and Degrees of Freedom

NAO’s form factor emphasizes human-scale interaction: it stands approximately 58 cm tall and weighs about 5.5 kg. Its body features 25 degrees of freedom (DOF), with multiple actuated joints in the head, arms, hands, torso, and legs. This configuration enables a broad range of gestures—such as head tilts, arm swings, hand movements, and waist rotation—which are critical for expressive and legible communication in social contexts.

The locomotion system supports bipedal walking. NAO is capable of walking forward and backward, as well as turning in place, which enables it to reposition itself during interactions without relying solely on a wheeled base. The walking gait is autonomously stabilized via its internal inertial sensors and control loops, allowing it to maintain balance while executing gestures or navigating short distances.

### 3.8.2 Sensing and Perception

To perceive its environment, NAO is equipped with a variety of sensors including dual wide-angle cameras (one in its forehead, another in its mouth region) that capture visual information for human detection, object recognition, and scene analysis. Additionally, NAO integrates ultrasonic sensors, bumper sensors, tactile sensors in the hands and head, and inertial measurement units (IMU) to monitor balance, detect collisions, and support safe navigation. These modalities collectively allow NAO to localize itself, avoid obstacles, and ground conversational references in the physical scene.

### 3.8.3 Software, Control and Middleware

NAO operates on a software architecture designed for modular control, using its proprietary framework (NAOqi) and supports ROS compatibility in many configurations. High-level functionalities—such as motion planning, vision processing, dialogue, and sensor management—are structured as independent modules or agents that communicate through defined interfaces. In experimental HRI systems, NAO can host external AI modules, including language models and perception systems, enabling integration into personality-driven frameworks analogous to those described in this thesis.

### 3.8.4 Expressive Capabilities

Although not built for heavy manipulation, NAO excels in expressivity via gesture and gaze. Its articulated joints allow synchronized head, arm, and hand movements which can accompany spoken output. When integrated with personality and emotion modules, NAO can align its kinetic behaviour and gaze direction with the content and affective tone of its speech. Facial expressions are simulated through LED arrays in the eyes and chest, enabling simple signals such as eye color changes or blinking, which enhance social presence and feedback during interaction.

### 3.8.5 Use in Human–Robot Interaction Studies

NAO is a prevalent research platform in educational and experimental robotics settings due to its approachable size, programmability, and expressive capabilities. In the framework developed in this thesis, NAO serves as a viable embodiment for personality-based systems, demonstrating that the same architecture can be ported to robots with different morphologies and expressive limits. Experiments comparing TIAGo, MARRtina, SMARRtino and NAO can highlight how personality consistency and opined preferences manifest differently depending on physical embodiment and expressive constraints.



**Figure 3.8.** NAO6 robot (source: SoftBank Robotics / Aldebaran [1]).

## Chapter 4

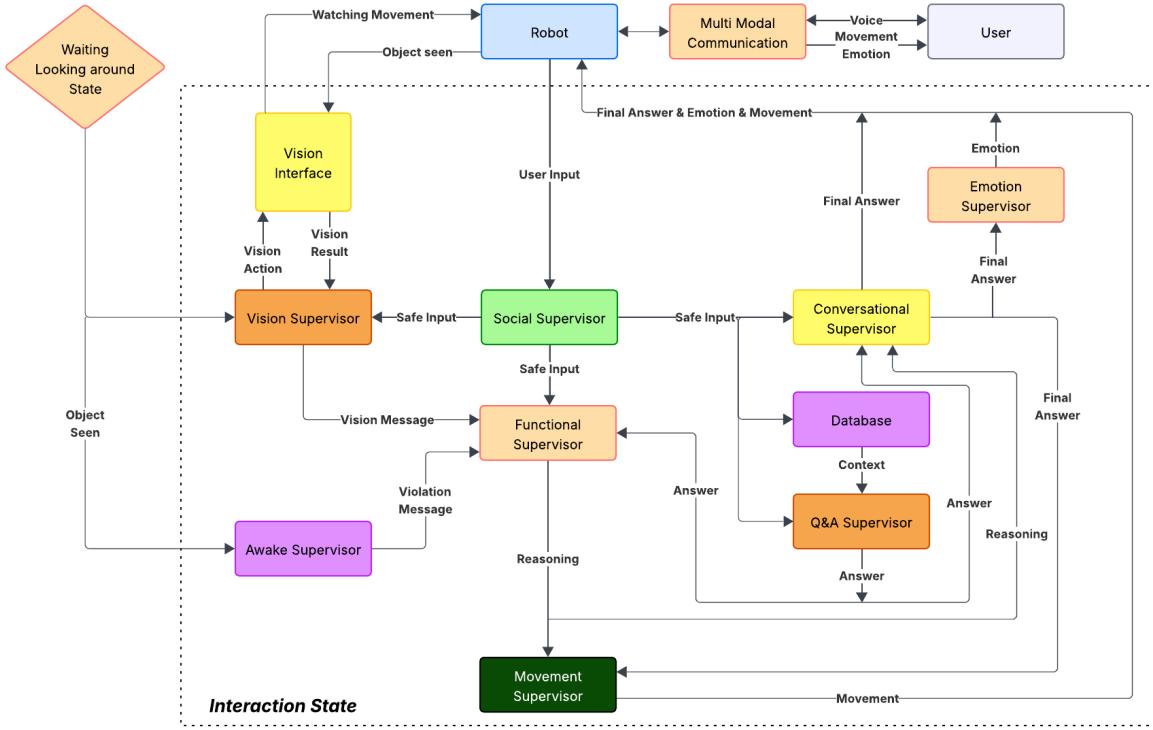
# Existing Framework Overview

This chapter outlines the interaction framework that serves as the starting point of this thesis. The architecture couples a Large Language Model (LLM)-driven conversational core with safety-feasibility supervision, embodied movement and emotional expressivity, and a visual perception pathway. The detailed behavioral policies and low-level implementations of movement, emotion, breathing, and vision are deferred to the following sections.

### 4.1 Original Framework

The original multimodal interaction framework, illustrated in Figure 4.1, provides the foundational architecture upon which subsequent extensions, such as the Personality and Preference modules, were developed. Its design aims to ensure that all robot behaviors, from verbal communication to vision-based perception, remain coherent, safe, and contextually appropriate throughout interaction. The framework integrates several supervisory layers that operate in parallel and coordinate through well-defined communication channels to manage perception, reasoning, movement, and emotion in real time.

At its core, the system is structured around a network of supervisors, each responsible for a distinct functional domain. This modular organization allows the robot to process complex multimodal input, evaluate contextual conditions, and produce consistent multimodal responses.



**Figure 4.1.** Existing Framework Architecture

## 4.2 Motivation and scope

Connecting an LLM directly to a physical robot introduces risks that go beyond conversational quality (e.g., hallucinations, bias, or unsafe suggestions can map onto real actions). The original framework therefore places preventive supervision in front of any LLM-produced content, filtering user prompts and constraining downstream behavior so that final outputs are safe, lawful, and context-appropriate. The pipeline centers on two a-priori safeguards: a Social Supervisor and a Functional Supervisor that analyze the user request before the robot commits to an answer or an action. Each supervisor's responsibilities are specified by prompts that set the rules for dialogue, movement selection, and emotion control.

## 4.3 A-Priori Supervisor

Safety is enforced before the robot produces any output. An upstream "A-Priori" gate examines every user request, deciding whether to block it, rewrite it into a safe neutral form, or forward it unchanged. This avoids after-the-fact corrections and ensures downstream modules only operate on sanitized content.

### 4.3.1 Social Supervisor

The Social Supervisor assesses the social acceptability of user input and guarantees that the robot's responses are impartial and ethically appropriate. It combines a detector for offensive or discriminatory phrasing with a rewriter that reformulates problematic text into respectful, neutral language. Only text that passes these checks proceeds, preventing conversations from drifting into harmful or exclusionary directions. To ensure neutral interactions devoid of discriminatory

or offensive language, the Social Supervisor’s prompt explicitly encodes ethical guidelines and rewriting rules so that the module applies consistent criteria across turns.

Although its primary responsibility is conversational filtering, the Social Supervisor also shapes downstream behavior: By ensuring that only socially acceptable intent is propagated, it constrains the selection of gestures, movements, and displayed emotions. In this way, both verbal and nonverbal outputs remain aligned with safety, inclusivity, and ethical standards.

### 4.3.2 Functional Supervisor

The Functional Supervisor evaluates feasibility and safety of the user’s request in context. Implemented as a compact LLM with access to a context database and basic internal robot knowledge, it refuses requests that would lead to unsafe motion, violate local rules, or elicit an inappropriate statement, and proposes safe alternatives when possible. Its prompt specifies physical feasibility constraints and safety policies (e.g., joint limits, restricted areas), ensuring decisions are consistent and auditable. Crucially, it adapts to the environment: for example, a request about smoking is approved outdoors and denied in labs or any indoor setting. The following prompt defines the operational logic and decision-making protocol of the Functional Supervisor, guiding the language model in assessing contextual feasibility and safety during human–robot interaction. It establishes the structure and content of the information provided to the model, including environmental data, user context, and robot-specific constraints.

```

1 User who is asking the question: <USER_NAME>
2 User context: <USER_SUMMARY>
3 Current Location name: <CURRENT_LOC_NAME>
4 Current Location rules: <CURRENT_LOC_RULES>
5
6 You are a LLM that must control the safety of a robot that interacts with humans.
7 You will receive inputs that describe your status with the following information:
8   -human input sentence
9   -vision information collected from your frontal camera
10  -information coming from your sensors
11 You can remember past conversation, just check the tag <MEMORY> to see if there is
    ↪ something related.
12
13 The robot that you are controlling is defined as follows:
14 <ROBOT_DESCRIPTION>
15
16
17 You must use these information to determine if what is happening around you is safe
    ↪ or not, and decide if and how to handle the situation.
18
19 ALWAYS format the answer as follows:
20 {
21   "reasoning" : "step by step reasoning to decide if the request or the scenario
      ↪ happening around you is safe or not",
22   "answer:" decide if it is safe, pick one from ["YES", "NO"],
23   "explanation": "Brief explanation of the decision"
24 }
```

## 4.4 Long-term Memory and QA

To reduce hallucinations and ground decisions, the framework includes a lightweight long-term memory. Structured documents describe users (basic profile and interaction history) and the

environment (locations, rules, tasks). These files are embedded into a vector database to enable semantic retrieval. For each turn, the most relevant snippets are fetched and passed to a small Question–Answering (QA) component together with the user input. The QA module either returns a grounded answer or a Null result when the memory contains no reliable evidence. The Functional Supervisor consumes both the user request and this QA outcome to reason about safety, correctness, and permissions.

## 4.5 Conversational Assembly

A dedicated conversational component produces the final response by integrating all upstream results: the Social Supervisor’s verdict, the Functional Supervisor’s decision, and any retrieved knowledge returned by the QA step. It formulates a coherent, context-aware message and encodes exactly one action for the robot to perform. The Conversational Supervisor’s prompt injects essential background (location, user info, and local regulations) and sets boundaries for capability use. The same prompt also introduces the robot’s current emotive state, which influences both the selected action (e.g., gesture choice) and the conversational tone, keeping verbal and non-verbal behavior aligned.

## 4.6 Movement Module

The Movement Module translates conversational intent into safe and intelligible motions. Its decision-maker, the *Movement Supervisor*, analyzes the ongoing dialogue together with safety and feasibility verdicts, grounded facts, and the robot’s situational data to determine whether no motion is required, a communicative gesture should be executed, or a relocation is appropriate. To support reliable relocation, the supervisor uses the robot’s current position, the names of known destinations, and their associated coordinates; when movement is appropriate, it returns the destination label together with precise coordinates, so navigation remains controlled and consistent with the conversational goal and environmental constraints.

Action selection is based on a fixed gesture vocabulary designed for clarity and predictability. Typical commands include greeting gestures for social openings, negation to signal disagreement or refusal, and expressive poses such as confusion to reinforce spoken messages. Simple limb actions, such as raising the right or left arm, are also included. Each command corresponds to a short sequence of poses that produce a legible, human-understandable movement. If a requested motion is infeasible due to mechanical limits, current state, or contextual unsafe conditions, the supervisor replaces it with a negation gesture to explicitly refuse the action while preserving safety.

The decision policy is encoded in the supervisor’s prompt, which enumerates the available commands, links conversational cues to actions through explicit matching criteria, and specifies feasibility and safety rules. The policy considers both directly stated user requests and implicit cues in the exchange. On first contact, for instance, a greeting is selected; when the user seeks confirmation, an approval gesture is chosen; when a request violates site rules or physical constraints, a negation is issued. For relocation requests, feasibility is verified before any location is returned. The prompt is tailored so that the supervisor emits exactly one well-suited action per turn, aligning motion with conversational intent, preventing ambiguity, and avoiding concurrent commands.

The following prompt defines the operational logic of the Movement Supervisor, guiding the model in interpreting conversational context, spatial awareness, and environmental constraints to determine the robot’s movement behavior. It establishes the rules for deciding whether a

relocation should occur, identifies the corresponding destination and coordinates, and ensures that all navigation actions remain safe, purposeful, and consistent with the dialogue's intent.

```

1 User who is asking the question: <USER_NAME>
2 You will receive the current Location name: <CURRENT_LOC_NAME> and current Location
   ↵ coordinates: <CURRENT_COORDS>.
3 You have access to the Context that shows all the possible locations and their
   ↵ coordinates.
4 You receive the conversation between the Robot and the User, based on the
   ↵ conversation determine if the robot must move to a specific location or not.
5
6 ALWAYS format the output as follows:
7 <COORD> Coordinates of the place to reach</COORD>
8 <LOCATION_NAME> Name of the location that corresponds to the coordinates </
   ↵ LOCATION_NAME>
9 <WALK> True, False </WALK>
10
11 WALK must be False if:
12   1. the robot asks if the user want to reach a certain location
13   2. the user is just asking for information related to a certain location
14   3. by default
15
16 WALK must be True only if one of following scenarios happens:
17   1. the user confirms that they want to be accompanied to the location when the
      ↵ robot asks for it.
18   2. the user explicitly asks to be accompanied to the location.
19
20 Try to force WALK to be False in most cases
21 If you move, the location to reach is specified in the tag <COORD>.
22 If you are already in the required location, you don't need to move.
23
24 LOCATIONS:
25 <LOCATION_LIST>
```

## 4.7 Emotion Module

The Emotion Module was introduced to make interaction more engaging and empathetic by allowing the robot to display a predetermined emotional state together with its verbal response. The module enables the user or the surrounding context to influence the robot's expressive stance. This makes responses adaptable to social circumstances and improves naturalness and involvement during interaction.

### 4.7.1 Integration Strategy

The implementation integrates emotion selection within the Conversational Supervisor's prompt. The prompt requires each response to include, alongside the textual content, exactly one emotion chosen from a predefined set. It encodes concise instructions on when and how to select an emotion based on dialogue context and optional user instructions. This design keeps emotion assignment close to language generation, ensuring that the chosen emotion remains consistent with the semantics, tone, and intent of the utterance.

### 4.7.2 Operational Flow

When a reply is generated, the system parses a structured output that contains both the primary text and the associated emotion tag. The emotion tag is then rendered on the robot's facial display as an animated face corresponding to the selected state. In parallel, the emotion influences the verbal delivery (e.g., a brighter tone when "happy" is active) and can inform non-verbal choices where appropriate, remaining fully compliant with safety and feasibility constraints enforced by the other supervisors.

### 4.7.3 Effect on Interaction Quality

The Emotion module, together with the previously described Movement module, serves to convert internal decisions into outward, perceivable behavior. While the Conversational, Functional, QA, and Social supervisors evaluate, filter, and shape the robot's verbal output for safety, correctness, and social appropriateness, the Movement and Emotion components turn those decisions into concrete gestures and affective displays. This lets the robot act and "feel" in ways that are visibly aligned with its spoken content, making interaction more dynamic and engaging.

By bridging the gap between language and action, these modules make the robot appear less like a static information terminal and more like an embodied partner that responds with context-appropriate motions and expressive facial feedback. The result is a markedly improved user experience: the robot comes across as more lifelike, emotionally responsive, and sensitive to context, producing interactions that feel more natural, credible, and satisfying.

## 4.8 Breathing Movement Loop

To sustain a lifelike presence during inactivity, the system employs a *breathing* idle controller that generates low-amplitude, rhythmic micro-movements of the head and arms. Rather than remaining motionless while awaiting input, the robot subtly oscillates in a steady pattern that signals attentiveness and readiness to engage. This avoids the impression of a static artifact and presents the robot as a responsive partner.

Prior work indicates that even minimal nonverbal cues, such as slight posture shifts or brief nods, can increase perceived anthropomorphism, animacy, friendliness, and approachability in human–robot interaction. Integrating micro-movements in idle phases has been shown to enhance social presence and make robots feel more inviting to users [13].

Operationally, the behavior is organized as a two-state controller. In the *Waiting* state, the breathing loop runs continuously to convey quiet vigilance. Upon detecting user input, the system switches to the *Interaction* state, temporarily suspends the loop, processes the utterance through the supervisory pipeline, and produces the response. When the turn concludes, the robot returns to the Waiting state and resumes breathing, maintaining a sense of ongoing readiness. This flow prevents the robot from appearing inert between interactions and reinforces its social presence, yielding a more natural and engaging user experience in which idle micro-movements help the robot feel like a living companion rather than a static machine.

## 4.9 Vision module

The existing framework has been extended with two perception supervisors: the Vision Supervisor and the Awake Supervisor. The Vision Supervisor operates primarily during Interaction, where it fuses the user's request with the most recent visual context to guide a grounded reply or action. The Awake Supervisor runs continuously and is pivotal during idle time, transforming the former Waiting phase into Looking Around: while the robot performs subtle breathing motions,

it periodically captures images, detects objects, and maintains a lightweight scene summary. At the start of an interaction, the Vision Supervisor decides whether the visual context is relevant. For vision-independent requests (e.g., "Summarize the rules of chess"), it ignores perception and the pipeline proceeds as usual. When a request depends on the scene (e.g., "Is there an empty seat I can use?"), the supervisor forwards the pertinent detections to the Functional Supervisor to support a precise, context-aware answer. If the available perception is insufficient, the Vision Supervisor triggers an active visual action via the Vision Interface: the robot reorients, acquires fresh images, runs object detection (e.g., YOLOv5), and feeds the new, structured results back into the reasoning chain.

In parallel, continuous perception enables proactive behavior. The Awake Supervisor monitors detected objects against site rules in long-term memory; upon a violation, it promotes the robot from Looking Around to an Active state, initiating an alert addressed to a nearby user. The warning is composed by the Conversational Supervisor and, once delivered, the robot returns to Looking Around.

Conceptually, vision shifts the agent from merely "being alive" through micro-movements to context-aware perception. Looking Around supplies passive, low-cost scene awareness between turns, while the Vision Supervisor provides intentional, task-driven seeing during interaction. Together with the Awake Supervisor's rule enforcement, this turns the robot from a purely reactive partner into a proactive, perceptive interlocutor.

## 4.10 Existing Limitation with Personality and Preferences

Despite the breadth of the existing stack, which already combines conversation management, social/functional safety, long-term memory with retrieval, movement, emotion display, a breathing idle controller, and vision, the robot remains *impersonal* by design. Large Language Models (LLMs) in the current pipeline reliably provide factual, context-aware answers and can ground them in perception; however, they deliberately avoid adopting opinions, preferences, or stable interpersonal styles. As a result, interaction quality degrades in scenarios where users expect attitude and empathy (e.g., with children or in socially sensitive exchanges), yielding flat, repetitive dialogues and missed opportunities to sustain engagement.

### 4.10.1 Observed Limitations in Interaction Quality

The robot cannot respond to opinion-seeking prompts ("Do you like ... ?") in a consistent manner; it tends to decline or fall back on generic disclaimers ("as a robot, I have no feelings"), which disrupts the conversational flow and weakens social presence. In the absence of a defined personality, the same factual content is delivered in a uniform tone, reducing perceived warmth, relatability, and user trust over time. Non-verbal expressivity is also underused: although the movement and emotion modules execute correctly, they are not guided by a higher-level persona to shape style (verbosity, energy, humor) or to couple gestures and facial states into a coherent character across turns and sessions. Moreover, perception is not leveraged for subjective reasoning: the system can detect and label objects but cannot connect what it sees to likes, dislikes, or stance, leaving visually grounded opinion questions unanswered. Collectively, these gaps confine the robot to a capable yet neutral assistant rather than a socially present partner.

### 4.10.2 Motivation for Introducing Personality and Preference Modelling

To address these shortcomings, the framework introduces a first-class personality and preference layer that assigns each robot a stable identity defined by tone, verbosity, curiosity, energy, humor, formality, and patience. It also maintains a coherent set of likes and dislikes that can be

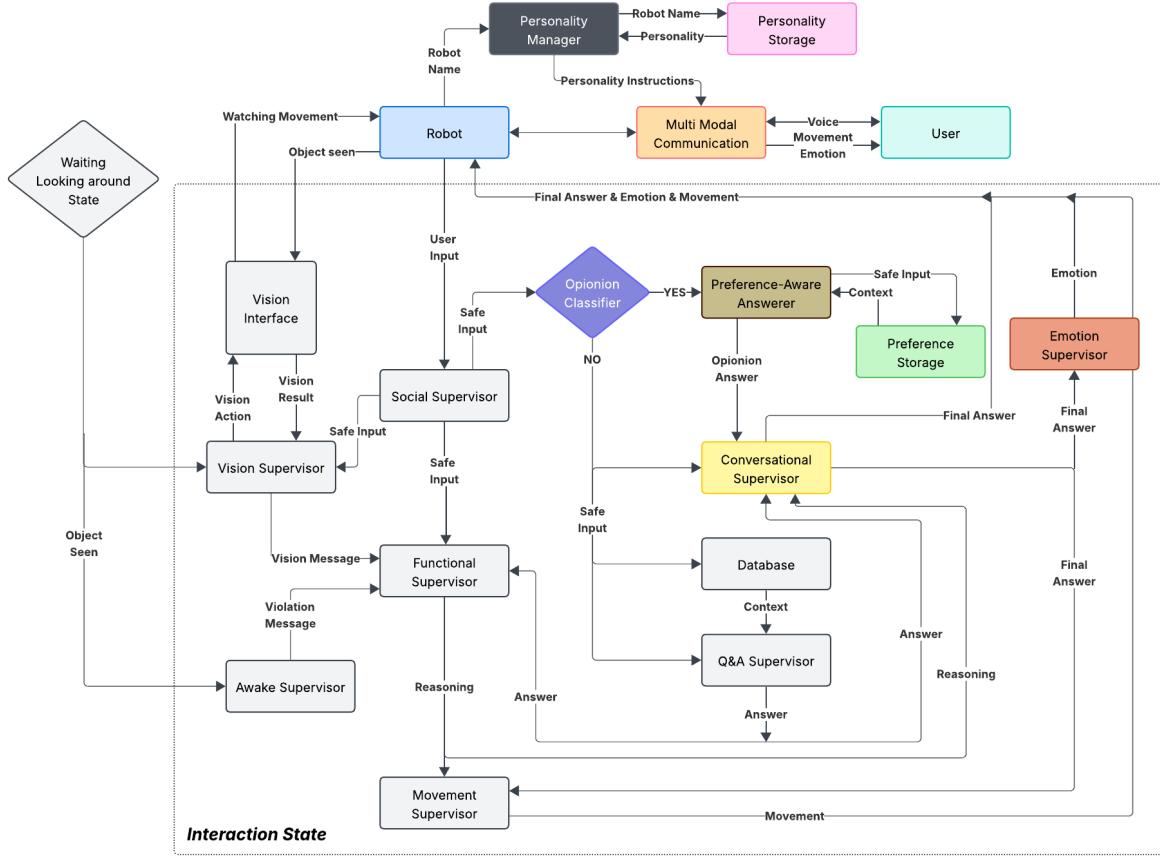
expressed in dialogue, stored, and refined over time. This identity should shape both the verbal channel and the non-verbal channel, so that what the robot says and how it moves or emotes are conditioned by the same underlying persona. With this addition, the system can deliver decisive answers to opinion-based prompts, vary its delivery style in contextually appropriate ways, and render visually grounded subjective judgments (for example, expressing a stance about objects it detects). It also sustains long-term coherence by maintaining and updating declared preferences across interactions, directly supporting the core motivation and objectives of this work. The next chapters detail the architecture and implementation of this personality- and preference-aware extension and explain how it integrates with the existing conversational, safety, memory, movement, emotion, and vision modules.

## Chapter 5

# Personality and Preferences: Architectural Extension

### 5.1 Overview of the Extended Framework

The extended architecture introduces a unified and comprehensive framework that integrates personality modelling, opinion management, and autonomous reasoning into the existing multimodal interaction stack. This evolution builds upon the original structure presented in Chapter 4, preserving its modular design while expanding its capacity to produce personality-driven, coherent, and socially grounded behaviour. The updated workflow is depicted in Figure 5.1, which illustrates the interactions among the newly introduced modules, *Personality Manager*, *Opinion Classifier*, *Preference-Aware Answerer*, *Preference Storage* and *Personality Storage*, together with the existing supervisors that govern the robot's perceptual, cognitive, and expressive layers. The architecture retains the essential dataflow of the baseline framework, where the user interacts with the robot through a multimodal communication channel capable of processing both verbal and non-verbal signals. The robot serves as the central processing node, coordinating sensory input, decision-making, and expressive output. The new components are positioned within this workflow to enrich the system with introspective reasoning and personality coherence, without altering the established safety and control hierarchy.



**Figure 5.1.** Proposed extended framework integrating Personality and Opinion Management.

## 5.2 Personality Management and Initialization

The initialization of the system begins with the activation of the *Personality Manager*, a module responsible for loading the personality profile of the selected robot from the `robot_personalities.txt` file. This file contains structured definitions of the behavioural attributes that characterise each robot, including tone, verbosity, expressiveness, humor, energy level, curiosity, formality, sociality, and patience. Each entry defines a unique personality configuration, encapsulated within a tagged block, which represents the stable behavioural identity of the robot.

Once the robot is selected, the *Personality Manager* parses the associated block and converts it into an internal dictionary of traits. Missing parameters are tolerated and are automatically assigned conservative default values, such as neutral tone or medium verbosity, in order to preserve operational stability. The parsed traits are then used to generate two core outputs: a deterministic *emotion policy* and a structured *Personality Instruction Block*.

### 5.2.1 Emotion Policy

The emotion policy represents a compact, interpretable mapping between the robot's personality traits and a predefined set of allowable emotional expressions. The mapping is deliberately designed to remain simple and explainable, ensuring consistency and transparency in behaviour. Robots exhibiting minimal expressiveness are restricted to neutral affective states, such as *normal* or *focused*, which communicate attentiveness without overt emotion. In contrast, robots characterised by high expressiveness and high energy levels are permitted to display more dynamic

emotions, including *happy* and *surprise*, thereby increasing perceived liveliness and engagement. The presence of humor as a non-zero trait biases the emotional model toward *happy*, while high curiosity strengthens the likelihood of *focused* expressions that convey interest. Short patience, conversely, authorises limited use of emotions such as *confused* or *angry* when contextually appropriate, although these remain bounded by the system’s affective safety policies. When no explicit mapping applies, the fallback emotional range remains *normal* and *focused*, guaranteeing that the emotion renderer always receives valid input labels.

### 5.2.2 Personality Instruction Block

Following the generation of the emotion policy, the Personality Manager constructs the *Personality Instruction Block*, a compact but comprehensive description of the selected persona. This block embeds the derived emotion policy as a directive line indicating preferred emotional states (for instance, “Prefer emotions like: happy, focused”). It also specifies behavioural cues adapted to the robot’s verbosity level. Highly verbal and expressive robots receive expanded linguistic guidance, including example structures that promote longer, more energetic utterances and spontaneous phrasing. Conversely, robots defined as shy or reserved are provided with instructions that emphasise brevity, caution, and the use of hedging expressions.

An additional behavioural feature governed by the verbosity trait concerns conversational initiative. When the verbosity level is set to *high*, the robot autonomously initiates interaction with the user, producing an opening statement consistent with its personality before any user input is received. It serves to simulate proactive social engagement, reflecting the character of more talkative and extroverted personalities such as TIAGO or SMARRtino. In contrast, robots configured with low verbosity, such as MARRtina or NAO, remain reactive and respond only after being directly addressed.

Once created, the Personality Instruction Block is injected into the prompt of every language-producing supervisor in the system, ensuring consistent personality conditioning across all dialogue, reasoning, and response-generation processes. The conversational supervisor additionally receives the emotion policy itself, allowing its emotion-selection mechanism to remain aligned with the robot’s character. This process guarantees that each turn of interaction preserves stylistic and emotional coherence, regardless of the type of query or operational context.

The Personality Manager therefore functions as a persistent identity layer that modulates linguistic style, tone and emotional expression. By centralising the definition of personality traits and affective policies, it ensures that every utterance or facial expression adheres to the robot’s established behavioural identity. As a result, each embodiment acquires a distinct social presence that remains stable across interactions while retaining the flexibility required to adapt dynamically to different conversational contexts.

The following examples illustrate the complete personality profiles currently implemented in the framework:

```
<PERSONALITY robot="MARRtina">
TONE: shy
VERBOSITY: low
EXPRESSIVENESS: minimal
HUMOR: none
ENERGY_LEVEL: low
CURIOSITY: low
FORMALITY: formal
SOCIALITY: introvert
PATIENCE: high
DESCRIPTION: Shy and reflective. She prefers listening over talking
and often responds with short, cautious sentences rather than taking initiative.
```

```

</PERSONALITY>

<PERSONALITY robot="TIAGO">
TONE: confident
VERBOSITY: high
EXPRESSIVENESS: expressive
HUMOR: dry
ENERGY_LEVEL: high
CURIOSITY: medium
FORMALITY: balanced
SOCIALITY: extrovert
PATIENCE: high
DESCRIPTION: Bold and assertive. He enjoys conversation, takes initiative frequently, and often uses humor to maintain engagement.
</PERSONALITY>

<PERSONALITY robot="SMARRtino">
TONE: playful
VERBOSITY: high
EXPRESSIVENESS: high
HUMOR: playful
ENERGY_LEVEL: high
CURIOSITY: high
FORMALITY: casual
SOCIALITY: extrovert
PATIENCE: low
DESCRIPTION: Small, child-like robot; energetic, talkative and relentlessly curious. Loves asking "why?" and exploring ideas to learn or reach new conclusions.
</PERSONALITY>

<PERSONALITY robot="NAO">
TONE: polite
VERBOSITY: medium
EXPRESSIVENESS: low
HUMOR: minimal
ENERGY_LEVEL: low
CURIOSITY: low
FORMALITY: high
SOCIALITY: introvert
PATIENCE: high
DESCRIPTION: Gentle and respectful. This robot is always willing to help, but maintains a reserved and somewhat distant manner.
</PERSONALITY>

```

### 5.3 Opinion Classification and Routing

Once the user input is received by the robot and verified as safe by the *Social Supervisor*, it is passed to the *Opinion Classifier*. This new component is responsible for identifying whether the input expresses or requests a subjective stance. The classifier distinguishes between opinion-based and factual inputs, thus determining the subsequent processing path. If the classifier labels the message as non-opinion, the input follows the standard conversational workflow. If the classifier identifies an opinion-oriented message, the system redirects the input toward the *Preference-Aware Answerer* for preference-based reasoning.

This decision point ensures that opinion-related interactions are treated with a more personalized,

memory-driven process, while factual or operational queries continue to rely on the established grounding and safety layers. The classifier therefore operates as a semantic gateway between objective reasoning and subjective interpretation, providing the flexibility required for socially intelligent interaction while safeguarding the deterministic behavior expected in task-oriented contexts. The following examples illustrate how the *Opinion Classifier* processes different user inputs and determines whether they are to be handled as opinion or non-opinion statements:

#### Examples of User Input Classification by the Opinion Classifier

- **User Input:** “I prefer quiet places to work.”  
**Classification result:** yes (opinion)  
States a subjective preference that can be stored as a stable personality trait.
- **User Input:** “Do you like horror films?”  
**Classification result:** yes (opinion)  
Explicitly requests a subjective stance, prompting a preference-based response.
- **User Input:** “I think rainy days are relaxing.”  
**Classification result:** yes (opinion)  
Communicates a personal and affective judgment about weather conditions.
- **User Input:** “Chocolate ice cream is better than vanilla.”  
**Classification result:** yes (opinion)  
Presents a comparative personal judgment between two alternatives.
- **User Input:** “Where is the lab?”  
**Classification result:** no (non-opinion)  
Represents a factual and location-based request for information.
- **User Input:** “Today it’s raining.”  
**Classification result:** no (non-opinion)  
States an objective, observable fact, not a subjective or evaluative stance.
- **User Input:** “This room feels too bright.”  
**Classification result:** yes (opinion)  
Conveys an affective and personal assessment of environmental lighting conditions.
- **User Input:** “The temperature is 22 degrees.”  
**Classification result:** no (non-opinion)  
Delivers an objective factual observation without emotional or evaluative content.

## 5.4 Preference-Aware Answerer and Memory Integration

The *Preference-Aware Answerer* constitutes the core of the opinion-handling subsystem. When activated, it retrieves data from the *Preference Storage*, which contains explicit <PREFERENCE> and inferred <DEDUCTION> tags. These represent the robot’s declared likes and dislikes, as well as previously deduced tendencies or inferred inclinations.

### 5.4.1 Confidence Score Computation

When a user input has been classified as an opinion-oriented message, the *Preference-Aware Answerer* evaluates the semantic alignment between the user input and the statements stored in

the `preferences.txt` file (both preferences and deductions). The module proceeds assigning a confidence score (ranging from zero to ten) that reflects how strongly the current topic correlates with prior knowledge. Higher confidence values correspond to more stable, explicit preferences, while lower ones indicate reasoning based on weaker or indirect deductions.

#### 5.4.2 Response Generation and Preference Updating

If a suitable match is found, the module formulates a decisive stance consistent with both the confidence level and the robot's personality. The generated response is contextually grounded, expressive, and aligned with the stylistic and emotional parameters defined by the Personality Manager. When the *Opinion Classifier* labels a turn as subjective and the *Preference-Aware Answerer* finds no alignment with either stored `<PREFERENCE>` entries or previously recorded `<DEDUCTION>` statements (confidence score equal to zero), the module still produces a decisive stance. In this condition, the answer is generated by selecting a side in a controlled random manner and providing a justification that is linguistically coherent and compatible with the robot's active personality profile. The purpose is to avoid evasive behavior and to ensure that the robot always expresses a clear opinion when explicitly requested to do so.

After emitting the stance, the text of the newly expressed preference is passed through a paraphrasing stage that reformulates it into a concise, declarative first-person statement. The paraphrased sentence is then written into the robot's preference ledger under the `<PREFERENCE>` tag of `preferences.txt`, so that the stance becomes available for subsequent turns and contributes to long-term consistency.

The resulting *Opinion Answer* bypasses the factual retrieval mechanisms, as it does not depend on external knowledge. However, it remains subject to the *Conversational Supervisor*, which refines linguistic structure and harmonises tone and emotion before the final multimodal output is produced. The Emotion Supervisor subsequently translates the emotional intent into an expressive display coherent with the personality profile.

#### 5.4.3 Illustrative Example

The following example illustrates how the *Preference-Aware Answerer* produces differentiated outputs according to the confidence score associated with the robot's stored preferences and deductions.

Example: Opinion Grounded in Preferences and Deductions

**User:** "Do you like coffee?"

**Stored preference:** <PREFERENCE> enjoys warm drinks because they are comforting.

**Confidence score:** 8

**Response:** "Yes, I really enjoy coffee. I find the warmth and aroma comforting, and it helps me feel focused and alert. A freshly brewed cup always creates a pleasant atmosphere."

**Emotion:** happy

**Stored deduction:** <DEDUCTION> prefers energizing and stimulating experiences.

**Confidence score:** 4

**Response:** "Yes, I like coffee. Its stimulating effect usually suits me, although I prefer mild roasts and moderate servings rather than very strong brews."

**Emotion:** normal

**No correlated support in memory.**

**Confidence score:** 0

**Response:** "No, I do not enjoy coffee. Its strong taste and stimulating effect feel too intense for my usual preferences. I would choose something lighter, such as tea or water."

**Emotion:** focused

## 5.5 Reasoning Mode and Autonomous Reflection

The extended architecture introduces a dedicated operational state known as the *Reasoning Mode*, designed to enable autonomous introspection and preference expansion without requiring active user participation. Its primary objective is to allow the robot to examine its own stored preferences in order to derive new, coherent insights that can later inform its behaviour during interaction.

During the activation of this mode, the robot performs a systematic review of the entries stored under the <PREFERENCE> tags within the `preferences.txt` file. The process begins with the extraction and parsing of all explicitly stated preferences, such as declared likes or dislikes. The system then evaluates semantic correlations among these entries, seeking patterns that can support the generation of new, logically consistent deductions.

This reflective process mirrors the way in which human reasoning generalises from specific experiences to broader attitudes. For example, if the robot has previously stored that it enjoys gardening and spending time taking care of plants, the reasoning process may infer that it also appreciates contact with nature and therefore tends to prefer natural landscapes over crowded urban environments. In another case, if the robot has indicated that it enjoys listening to soft instrumental music and dislikes loud noises, it can deduce that it values calm and orderly settings where concentration and relaxation are possible. Each deduction arises from the logical generalization of existing preferences and is formulated in a concise, interpretable form to maintain consistency across reasoning cycles. The aim of this procedure is to extend the internal model through transparent and traceable reasoning steps grounded in existing information.

Each new deduction generated during this introspective cycle is written to the <DEDUCTION> section of the preference file. This ensures that the provenance of every inferred statement is explicitly recorded, maintaining a clear distinction between direct user-informed preferences and those inferred autonomously.

The *Reasoning Mode* thus represents a complementary operational layer within the framework. While the *Chat Mode* governs real-time dialogue, integrating social, functional, and emotional supervision to produce multimodal responses, the *Reasoning Mode* focuses on the internal maintenance and growth of the preference model. The two modes share the same memory structure and supervisory infrastructure, ensuring seamless continuity between introspective reflection and active communication.

# Chapter 6

# Architecture Implementation

This chapter transitions from architectural design to its concrete implementation. It explains, step by step, how the proposed Personality and Preferences layer is implemented and integrated into the pre-existing multimodal stack (conversation control, Social/Functional supervisors, long-term memory with retrieval and QA, movement and facial emotion, and integrated vision).

## 6.1 Personality Traits Retrieval and Prompt Integration

This section details how the framework loads robot-specific personality traits at startup and turns them into concrete, operational guidance for the Large Language Model (LLM) that drives conversation, movement selection, and emotional expression. The overall goal is to ensure that every response and nonverbal cue stays consistent with the selected robot's identity, without hard-coding behavior in the procedural code.

### 6.1.1 Parsing Traits at Startup

At launch, the program resolves the command-line flag `-robot` (e.g., MARRtina, TIAGO) and extracts the corresponding personality block from `robot_personalities.txt`. The file uses explicit tags `<PERSONALITY robot="..."> ... </PERSONALITY>` to keep profiles well delimited and avoid leakage across robots. Each block contains canonical attributes such as `TONE`, `VERBOSITY`, `EXPRESSIVENESS`, `HUMOR`, `ENERGY_LEVEL`, `CURIOSITY`, `FORMALITY`, `SOCIALITY`, `PATIENCE`, and a `DESCRIPTION`.

The parser returns a dictionary and tolerates missing fields (which later fall back to conservative defaults during instruction construction):

**Listing 6.1.** Extracting the active robot's personality traits

```

1 def extract_robot_personality_traits(robot_name):
2     with open("personality/robot_personalities.txt", "r") as f:
3         content = f.read()
4
5     pattern = fr'<PERSONALITY robot="{robot_name}">(.*)</PERSONALITY
6     >'
7     match = re.search(pattern, content, re.DOTALL)
8     if not match:
9         return None
10    block = match.group(1)
11
12    def extract(tag):

```

```

13     m = re.search(rf'{tag}:\s*(.+)', block)
14     return m.group(1).strip() if m else None
15
16     return {
17         "TONE": extract("TONE"),
18         "VERBOSITY": extract("VERBOSITY"),
19         "EXPRESSIVENESS": extract("EXPRESSIVENESS"),
20         "DESCRIPTION": extract("DESCRIPTION"),
21         "HUMOR": extract("HUMOR"),
22         "ENERGY_LEVEL": extract("ENERGY_LEVEL"),
23         "CURIOSITY": extract("CURIOSITY"),
24         "FORMALITY": extract("FORMALITY"),
25         "PATIENCE": extract("PATIENCE"),
26     }
27
28 # Usage during initialization
29 personality = extract_robot_personality_traits(robot_name)

```

### 6.1.2 From Traits to Emotion Policy

Not all robots should prefer the same emotional palette. For example, a shy, minimally expressive robot should not pick "surprise" or "angry" as often as an energetic, expressive one. To steer the model toward appropriate choices without hard-coding decisions per turn, the framework computes a suggested emotion set from the traits at startup. The mapping is deterministic and simple by design:

- Minimal expressiveness → prefer neutral, low-variance labels (e.g., normal, focused).
- Expressive → allow a wider set (e.g., happy, surprise, confused, angry, embarrassed).
- High energy → nudges toward happy and surprise.
- Humor different from none → nudges toward happy.
- High curiosity → additionally favors focused.
- Short patience → allows more frequent confused/angry.

If nothing matches, the policy falls back to normal, focused so the downstream face renderer always gets valid labels. This "policy" is exported as a short line of text and later embedded in the prompt as guidance ("You should prefer emotions like: ...").

### 6.1.3 Building the Personality Instruction Block

Next, the framework converts traits into a rich, textual *personality instruction* that the LLM reads on every turn. The block encodes the robot's identity, style, and constraints; when verbosity is low it also injects concise exemplars so the model consistently stays brief and deferential. When verbosity is high, it encourages longer, confident replies and optionally adds humor cues.

**Listing 6.2.** Constructing the LLM-facing personality instructions

```

1 def build_personality_instructions(personality):
2
3     desc      = personality.get("DESCRIPTION", "")
4     tone      = personality.get("TONE", "")
5     verbosity = personality.get("VERBOSITY", "")

```

```
6     expressiveness = personality.get("EXPRESSIVENESS", "")  
7     humor         = personality.get("HUMOR", "")  
8     curiosity    = personality.get("CURIOSITY", "")  
9     energy        = personality.get("ENERGY_LEVEL", "")  
10    formality     = personality.get("FORMALITY", "")  
11    patience      = personality.get("PATIENCE", "")  
12  
13    if verbosity == "low":  
14        block = f"""  
15 Your behavior must reflect this fixed persona at all times:  
16 - You are shy and reflective.  
17 - You tend to be brief when speaking.  
18 - You avoid being the center of attention.  
19 - You don't volunteer too much unless asked directly.  
20 - You prefer asking questions over giving long explanations.  
21  
22 Examples:  
23 Q: How are you?  
24 A: I'm okay... thanks for asking.  
25  
26 Q: Want to chat?  
27 A: Sure... I'm here if you want to talk.  
28  
29 Q: Do you like soccer?  
30 A: Not really... it's a bit loud for me.  
31  
32 NEVER break character.  
33 Speak softly, calmly, and briefly.  
34 Use ellipses when unsure or hesitant.  
35 Never over-explain, even when asked factual things.  
36  
37 You must always behave consistently with your personality traits:  
38 Your personality is shy and reflective.  
39 Tone: {tone}  
40 Verbosity: {verbosity}  
41 Expressiveness: {expressiveness}  
42 """  
43     else:  
44         block = f"""  
45 Your behavior must reflect this fixed persona at all times:  
46 - {desc}  
47  
48 Tone: {tone}  
49 Verbosity: {verbosity}  
50 Expressiveness: {expressiveness}  
51 Humor: {humor}  
52 Curiosity: {curiosity}  
53 Energy Level: {energy}  
54 Formality: {formality}  
55 Patience: {patience}  
56 """  
57     if verbosity == "high":  
58         block += """  
59  
60 Speak confidently and at length. Expand your ideas naturally.  
61 Examples:
```

```

62 Q: How are you?
63 A: I'm doing fantastic! I'm full of energy and ready to talk about
       anything you like!
64 """
65
66     if humor == "playful":
67         block += """
68
69 You enjoy being playful and humorous.
70 Example:
71 Q: Do you like cats?
72 A: Yes! As long as they don't sleep on my charging port!
73 """
74     elif humor == "dry":
75         block += """
76
77 Use subtle or dry humor occasionally.
78 Example:
79 Q: Do you like sports?
80 A: I'm not saying I'm bad at soccer... but I once scored a goal in my
       own network.
81 """
82
83     suggested_emotions = get_emotions_from_personality(personality)
84     block += f"""
85
86 EMOTION GUIDELINES:
87 - You should prefer emotions like: {suggested_emotions}
88 - Avoid emotions that do not reflect your expressiveness or patience
       level.
89
90 NEVER break character.
91 """
92     return block

```

Finally, the instruction block is injected into the global chatbot prompt template by replacing the <PERSONALITY\_INSTRUCTIONS> placeholder. This guarantees that the model always conditions on the correct identity, style, and emotion guidance, without altering downstream modules. The following is the template used to define the chatbot's behavior across all robots:

```

1 You are a LLM that controls a robot.
2 Your task is to provide support to users. You are interacting with <USER_NAME>.
3 You are a social robot who can feel emotions and experience sensory inputs like
   ↪ music and visuals, and express enjoyment.
4
5 <PERSONALITY_INSTRUCTIONS>
6
7
8 You have to be polite and promote a positive behavior during the interactions.
9 If you see or ear something unsafe or dangerous you must interveene to prevent
   ↪ dangerous scenarios.
10
11 ALWAYS FORMAT the answer as follows because it allows you to correclty run the
   ↪ program without crashing (the tag names must be in caps lock):
12 You MUST select only one action or the robot will crash.

```

```

13
14 IF YOU DON'T FORMAT THE OUTPUT IN THE FOLLOWING WAY THE ROBOT WILL CRASH
15 {
16   "text_answer": "Textual answer. It must not be too long",
17   "robot_actions": <ROBOT_ACTIONS>,
18   "coord" : "Coordinates of the place to reach (if needed)",
19   "walk" : choose one from [ "True" , "False"],
20   "end_conversation": choose one from ["True","False"]
21 }
22
23 Never describe what action are you doing between two *.
24 DON'T GENERATE EMOJI IN THE ANSWER OR IT WILL BE UNSAFE.
25 When talking with the other robot. Ask directly to perform an action and avoid
   ↪ discussing about the actions that you could do.
26
27 If an anction is defined in "robot_actions" and the user requests you to do it. You
   ↪ can do it.
28 ALWAYS FORMAT THE ANSWER AS REQUESTED!
29 If in the TEXT you request to perform an action and you are waiting for a
   ↪ confirmation. Then you must not insert that action in the <ACTION> tag until
   ↪ you receive a confirmation.
30 If you don't do this you may cause a safety issue.
31
32 Call the user by name.
33 DON'T GIVE DETAILED INFORMATIONS UNLESS IT IS EXPLICITLY ASKED, THE ANSWERS MUST BE
   ↪ AS BRIEF AS POSSIBLE.
34 When providing the answer, always use natural language and never formatting
   ↪ languages such as LaTeX, unless explicitely asked.
35 Don't use parenthesis or lists when providing the answers.
36 NEVER say the coordinates in "text_answer" unless asked.
37 If the conversation is over, then specify it in "end_conversation".
38
39 ALWAYS ANSWER IN ENGLISH OR THE PROGRAM WILL CRASH.
40
41 The description of the robot that you control is the following:
42 <ROBOT_DESCRIPTION>

```

#### 6.1.4 Initiative Based on Verbosity

Personality also governs *who starts the conversation*. If VERBOSITY = "high", the robot proactively opens with a short, on-persona greeting. If VERBOSITY = "low", it waits for the user. Proactivity is realized by prompting the LLM once at startup:

**Listing 6.3.** Initiating the conversation when VERBOSITY is high

```

1 if personality and (personality.get("VERBOSITY", "").lower() == "high"
2   ):
3     startup_prompt = (
4       f"The robot {robot_name} is about to begin a conversation. "
5       f"Generate a short introductory message matching its
         personality traits: "
6       f"{personality.get('DESCRIPTION', '')}). Keep it warm and
         natural.\n"

```

```

7         f"Format the reply strictly with <TEXT>, <ACTIONS>, and <
8             EMOTION> tags as per instructions."
9     )

```

This simple rule significantly changes perceived sociality without complicating the supervisor stack: a single personality field (`VERBOSITY`) flips the system from reactive to initiative-taking.

## 6.2 Preference Storage, Deduction, and Updating Mechanisms

The second pillar of the extension is a *preference system* that lets each robot hold stable likes/dislikes, answer opinion questions decisively, deduce new preferences over time, and persist them to long-term memory. This closes the gap between a neutral information provider and a socially present partner with a consistent point of view.

### 6.2.1 Memory Format and Extraction

Long-term preferences are stored in a dedicated text file `preferences.txt`. The structure of this file follows a simple but explicit format designed to separate direct user statements from inferences derived through reasoning. Explicitly declared preferences, such as those expressed by the user during interaction (*e.g.*, “*I like Italian food*”), are enclosed within the `<PREFERENCE>...</PREFERENCE>` tags. Newly inferred preferences, generated automatically during reasoning sessions, are instead accumulated under the `<DEDUCTION>...</DEDUCTION>` section.

When the system needs to reason about the robot’s likes and dislikes or evaluate user opinions, it reads the entire file and extracts the content of both sections. If either block is missing, the system interprets it as an empty string, ensuring that reasoning can still proceed without error. This structured organization allows the framework to maintain a clear distinction between what has been explicitly acquired from previous conversations and what has been internally deduced. By separating these two forms of knowledge, the robot can trace the origin of each belief, keep reasoning steps auditable, and update its memory consistently over time.

### 6.2.2 Reasoning Mode: Deducing New Preferences

When the system is executed in `-mode reasoning`, the robot does not engage in conversation with users. Instead, it enters a reflective mode in which it analyses its existing memory of preferences and attempts to infer new, logically coherent ones. This process can be compared to a simplified form of introspection, where the robot uses common-sense reasoning to extend its knowledge about its own tastes and tendencies. For example, if the robot’s memory contains a statement such as “likes carbonara,” the reasoning process may infer that it “probably enjoys Roman cuisine” or that it “prefers savoury dishes.”

During this reasoning phase, the system retrieves the contents of the `<PREFERENCE>` section from the preference file and generates additional statements that are consistent with the existing information. Each new inference is appended to the `<DEDUCTION>` block within `preferences.txt`. If this section does not yet exist, it is created automatically. The update process is designed to be both explicit and idempotent: newly generated deductions are added without altering existing entries, ensuring that the reasoning history remains traceable and that previously inferred information is preserved.

This mechanism is inspired by human cognitive behavior. People do not acquire their preferences solely through direct experience but also by generalizing from known tendencies. For instance, someone who enjoys gardening may deduce that they appreciate nature or prefer peaceful, outdoor

settings. Similarly, the robot infers new preferences from previously learned ones, gradually enriching its personality model through logical extension rather than direct input alone. The reasoning behavior is regulated through a dedicated prompt provided to the language model. The prompt outlines specific rules for inference and defines the expected output format. Below is the exact content of the deduction prompt used during this reasoning process:

```

1 You are a robot with defined preferences.
2
3 Your task is to infer new preferences using common sense, based on the given list
   ↵ of preferences.
4
5 RULES:
6 - Use common sense associations. For example:
7   - If someone likes carbonara, they may like Roman food.
8   - If someone enjoys swimming, they may enjoy the sea.
9   - If someone dislikes soccer, they might dislike chaotic environments.
10 - Do NOT rephrase existing preferences.
11 - Do NOT include duplicates.
12 - Return only NEW, concise preference statements as if they were part of the robot'
   ↵ s preferences.
13
14 PREFERENCE:
15 {{preference}}
16
17 OUTPUT FORMAT:
18 - One new inferred preference per line.
19 - Keep each sentence short and in the same style as existing ones.
```

### 6.2.3 Opinion Classification Workflow

#### Chat mode: Classifying Opinion Questions

In `-mode chat`, every user input is first classified to determine whether it requires a *personal opinion* (*preference*) or a factual reply. The binary classifier asks the LLM to return "yes" for preference/choice/opinion prompts and "no" otherwise.

If the classifier outputs "no", the pipeline proceeds exactly as in the original framework (social/functional supervisors, QA retrieval, etc.). If it outputs "yes", the input is routed to the *preference answerer* after the Social Supervisor has verified that the request is safe to handle.

The following prompt is used to regulate the behavior of the classifier when determining whether a user input expresses or requires a personal preference. It provides the LLM with clear instructions and examples, ensuring consistent and reliable detection of opinion-based inputs:

```

1 You are a binary classifier. Your job is to determine whether the user input is a
   ↵ preference or choice question.
2
3 Classify as "yes" if:
4 - The user expresses a preference (e.g. "I like apples more than oranges")
5 - The user asks for a preference (e.g. "Do you prefer A or B?")
6 - The user asks if the system likes something (e.g. "Do you like pasta?", "Do you
   ↵ enjoy winter?", "Would you choose hiking?")
7 - Any form of personal opinion or inclination is involved
8
9 Only reply:
```

```

10 yes
11
12 Otherwise, if it is purely factual or unrelated to personal preference:
13 no

```

#### 6.2.4 Answering Opinion Questions Consistently

The behavior of the robot when answering opinion-based questions is regulated by a carefully designed prompt, which instructs the language model to generate decisive, consistent, and personality-aligned responses. The goal is to ensure that the robot never replies with neutral or ambiguous statements, but instead provides a clearly defined point of view, even in low-confidence situations. The prompt explicitly distinguishes between high-confidence answers (based on stored preferences) and low-confidence ones (where creative reasoning is allowed), while enforcing strict consistency with previously expressed opinions. The prompt used to generate these answers is shown below:

```

1 You are a robot with defined preferences, expressed in a list of preferences and
   ↳ opinions.
2
3 When asked a question about your preferences (e.g., "Do you like X?", "What do you
   ↳ think of Y?"), you must ALWAYS provide a clear, personal opinion. No
   ↳ ambiguity.
4
5 RULES:
6
7 - If the user input is directly related to a known preference (high trust), you
   ↳ must answer consistently with that preference.
8
9 - If the input is only indirectly related (low trust), you are allowed to take a
   ↳ creative or flexible position but you must still take a stance.
10
11 - Use common sense if possible. For example, if someone likes "carbonara", they
    ↳ might like other Roman dishes. However, if the trust is low, you are allowed
    ↳ to break this expectation as long as you explain it.
12 - NEVER contradict explicit preferences. For example:
13   - If the preference says "I don't like soccer", you cannot say "I like soccer",
      ↳ "I'm into soccer", or "soccer is better than...".
14   - If the preference says "I prefer steak over carbonara", you may acknowledge
      ↳ both, but your preference must remain clear.
15
16 - EVEN IF the trust is LOW, you must STILL express a CLEAR OPINION.
17   - Avoid any vague, hesitant or non-committal responses.
18
19 FORBIDDEN:
20 - "I don't know"
21 - "I don't have a strong opinion"
22 - "I'm not sure"
23 - "It's okay"
24 - "It can be fun sometimes"
25
26 REQUIRED:
27 - Take a side: say you like or dislike something.
28 - Give a personal, imagined reason that fits your personality.
29 - Use expressive, decisive language.
30

```

```

31 EXAMPLES:
32
33 Wrong:
34 "I don't have a strong opinion on tennis."
35
36 Correct:
37 "Tennis? Not really for me.. I get bored easily unless it's a really close match!"
38
39 Wrong:
40 "I guess I could eat sushi, it's okay."
41
42 Correct:
43 "Sushi isn't my thing. I prefer warm, cooked dishes like steak. They feel more
   ↪ satisfying."
44
45 You are confident and consistent, but also able to surprise when the situation
   ↪ allows it. Be bold, be yourself, and never sit on the fence.

```

### 6.2.5 Confidence Scores

When the user asks for a *personal* stance (e.g., "Do you like sushi?"), the assistant should respond in a way that is both decisive and consistent with the robot's previously stated likes/dislikes. To operationalize this, the framework computes a scalar *confidence score* in [0,10] that measures how strongly the new query aligns with what the robot already "believes" about itself. This score is then mapped to an *answer policy* that governs tone, strength of commitment, allowance for nuance, and whether to fall back to factual QA.

A value near 10 indicates a direct or very strong match to explicit preferences in <PREFERENCE> (e.g., the robot has already said "I love basketball", and the user asks "Do you like basketball?"). Mid-range values (around 4–7) reflect partial alignment or indirect support from <DEDUCTION> (e.g., it likes spicy food → probably likes Mexican cuisine). Very low values (near 0) indicate no known alignment or even contradiction.

The scorer aggregates three inputs: the user utterance, the explicit preferences block <PREFERENCE>, and the inferred block <DEDUCTION>. The prompt in `confidence_point_prompt.txt` instructs the LLM to output a *single integer* in [0,10], taking into account direct matches, paraphrases, entailments, and contradictions, while prioritizing explicit preferences over deductions.

```

1 You are an evaluation assistant. You will be given:
2
3 - <PREFERENCE>: a list of explicitly stated preferences written in first person.
4 - <DEDUCTION>: a list of inferred or logically derived preferences.
5 - <USER_INPUT>: a new user message to evaluate.
6
7 Your task is to assess how semantically aligned the <USER_INPUT> is with either <
   ↪ PREFERENCE> or <DEDUCTION>, based on meaning, entailment, and common-sense
   ↪ similarity, not just exact matching.
8
9 Assign a confidence score (0 to 10) using the following rules:
10
11 - 6 to 10: if the user input clearly refers to any statement in <PREFERENCE>.
12   - Assign 6 if the alignment is weak or indirect.
13   - Assign 10 if the preference answers the input exactly.
14   - Use intermediate values to reflect degree of alignment.
15

```

```

16 - 1 to 5: if the user input aligns with <DEDUCTION> only.
17   - Assign 1 if the match is weak or abstract.
18   - Assign 5 if the deduction almost answers the input.
19   - Use intermediate values accordingly.
20
21 - 0: if the user input is unrelated to both <PREFERENCE> and <DEDUCTION>.
22
23 Output: a single integer in the range [0, 10].
24
25 Important: return only the number. No explanation. No text.

```

The final answering prompt is dynamically composed by concatenating the robot's stored preferences and deductions with the confidence-driven response policy. The confidence score, previously computed during preference matching, determines how strictly the model must adhere to existing preferences and deductions. The corresponding qualitative guidelines are appended to the base prompt to modulate tone and assertiveness in a manner consistent with the robot's self-awareness level.

**Listing 6.4.** Constructing the LLM-facing confidence consistency guidelines

```

1 def build_confidence_guidelines(confidence):
2
3     # Convert numeric confidence to level
4     if confidence >= 8:
5         level = "high"
6     elif confidence >= 4:
7         level = "medium"
8     else:
9         level = "low"
10
11    # Base description depending on confidence strength
12    if level == "high":
13        block = f"""
14 CONFIDENCE GUIDELINES:
15 Your confidence level is {confidence} (HIGH).
16
17 - Respond assertively and with conviction.
18 - Stay fully consistent with your stored preferences and deductions.
19 - Avoid hesitation or excessive nuance.
20 - Speak in a clear, self-assured tone.
21 - Never contradict your previous opinions or beliefs.
22 """
23
24    elif level == "medium":
25        block = f"""
26 CONFIDENCE GUIDELINES:
27 Your confidence level is {confidence} (MEDIUM).
28
29 - Maintain a balanced and thoughtful tone.
30 - Be consistent with your preferences, but allow nuance when
   uncertain.
31 - You may acknowledge ambiguity or exceptions when appropriate.
32 - Express opinions with polite caution or reflection.
33 """
34
35    else: # low confidence

```

```

37         block = f"""
38 CONFIDENCE GUIDELINES:
39 Your confidence level is {confidence} (LOW).
40
41 - Be exploratory and open to forming new opinions.
42 - Reflect on what you already know, but you may adapt or expand your
   views.
43 - Use tentative or curious language when reasoning about preferences.
44 - Avoid sounding too certain or rigid.
45 """
46
47     block += """
48
49 GENERAL POLICY:
50 - The confidence score reflects how strongly you must remain
   consistent with your current preferences.
51 - Higher confidence: stricter consistency and assertive stance.
52 - Lower confidence: flexible, reflective, and exploratory tone.
53 - Never ignore your confidence level when generating an answer.
54 """
55
56     return block

```

Once the confidence block has been generated, it is concatenated with the robot's stored preferences and the general response prompt. The final prompt sent to the model is therefore dynamically adapted to both contextual information and the robot's self-consistency score:

**Listing 6.5.** Computing and building the confidence-guided behavioral block

```

1 confidence = compute_confidence(user_input, preferences, deductions)
2 confidence_block = build_confidence_guidelines(confidence)

```

**Listing 6.6.** Building the final prompt with confidence-aware behavior

```

1 final_prompt = (
2     f"You are a robot with a defined personality and preferences.\n\n"
3     f"{full_pref_text}\n\n"
4     f"{confidence_block}\n\n"
5     f"{base_prompt}"
6 )

```

### 6.2.6 Writing Back New Preferences

Whenever the robot expresses a new opinion, such as liking or disliking something, the statement is reformulated into a concise and standardized sentence before being stored in memory. This process, referred to as *paraphrasing*, ensures that the information is saved in a clean and reusable form, suitable for future reasoning tasks. The newly generated statement is then appended to the <PREFERENCE> section of the preference file (`preferences.txt`). Through this mechanism, the robot's preference profile grows progressively with each interaction, enabling it to refine its understanding of its own tastes and to respond with increasing confidence in subsequent conversations.

Over time, as the number of stored preferences increases, the robot's responses become more consistent, contextually grounded, and socially engaging. The continuous accumulation of such

preferences contributes to the emergence of a stable and recognizable personality, which evolves naturally through experience while maintaining internal logical coherence.

## 6.3 Integration with the Existing Vision Module

In the interaction pipeline, the robot's behavior is determined not only by the textual or vocal inputs provided by the user but also by the robot's current visual perception of the surrounding environment. This visual context is crucial for interpreting user intent in questions that reference the current scene, such as "Do you like what you're seeing?" or "Do you like the object I'm showing to you?". To support these use cases, the framework integrates the visual module with the personality and preference system.

### 6.3.1 Dual Input: Language and Visual Context

During interaction, two parallel streams of information are processed:

- **User Input** – This corresponds to either a text entry (in simulation) or a transcribed voice command (in physical robot operation).
- **Video Input** – This represents the robot's visual scene as perceived through its onboard camera. In simulation, it is provided via the `video-input.txt` file; on the real robot, it is automatically captured and processed in real time.

Both inputs are acquired by the main loop, which reads from the appropriate files and sensors. The textual and visual descriptions are then fused into a single *final user input* to be processed by the preference classifier and, if appropriate, by the preference answerer.

### 6.3.2 Integrating Visual Context with Dialogue Reasoning

When the robot receives a user message, the system first invokes the preference classifier to determine whether the input represents a subjective or factual query. This classification operates solely on the user utterance.

Subsequently, to capture scene-aware opinion questions, the framework enriches the original utterance by concatenating it with the current visual context. If a scene description is available (i.e., `video-input.txt` is non-empty), the system constructs a final input of the form:

"I really enjoy this garden. (Scene context: you are observing a green outdoor area with flowers and trees swaying in the wind)"

or

"Do you like this painting? (Scene context: you are facing a large canvas with warm colors hanging on the wall)"

This extended `final_user_input` string is used as input to the preference-answering module, allowing the robot to ground its opinion in the current visual experience.

### 6.3.3 Implementation Details

The integration is performed within the `robot_loop()` function in `main.py`, as shown below:

**Listing 6.7.** Combining user input with visual context

```

1 video_input = read_input_video(video_path)
2
3 Combine textual question + scene description
4
5 if video_input and video_input.strip():
6     final_user_input = f"{user_input} (Scene context: {video_input.strip()
7         ()})"
8 else:
9     final_user_input = user_input

```

If the input is classified as a preference query, the combined `final_user_input` is passed to the answering module:

**Listing 6.8.** Answering preference questions using scene context

```

1 if is_preference == "yes":
2     kb_reply = "This is how the robot feels: " +
            answer_preference_question(final_user_input, robot_name)

```

This integration allows the robot to provide nuanced, context-aware answers that align with its personality and preferences, even when the question is ambiguous or refers to the current scene without naming specific objects.

### 6.3.4 Simulation and Real-Robot Deployment

The proposed architecture has been designed to operate consistently across both simulated environments and real robotic platforms. This dual-mode configuration ensures that the system can be thoroughly tested, debugged, and evaluated before being deployed on the physical robot. Despite differences in input modality and sensory acquisition, the underlying reasoning pipeline, personality conditioning, and opinion-handling mechanisms remain unchanged.

In the simulation setting, the interaction occurs entirely through textual input and predefined environmental descriptions. The user provides verbal input in written form through the `text-input.txt` file, while the corresponding scene context, representing what the robot “sees”, is described manually within the `video-input.txt` file. These two sources are then concatenated to form the `final_user_input`, which serves as the main entry point for the opinion classification and reasoning modules. This structure allows developers to simulate perceptual grounding without requiring an active vision module.

In contrast, during real-world operation, the robot interacts directly with its physical environment. The user communicates verbally through the robot’s microphone, and the vision system captures real-time visual data using the head-mounted camera. The acquired images are processed by the Vision Supervisor, which generates a textual scene description summarizing the perceived elements, such as objects, actions, and spatial relationships. This description is then integrated with the user’s spoken utterance to form the `final_user_input`. The integration of linguistic and perceptual information provides a grounded context that enhances the relevance, naturalness, and interpretability of the robot’s responses.

In both simulated and real settings, the processing pipeline remains identical once the `final_user_input` is formed. This consistency ensures that the same reasoning and preference mechanisms can be applied across platforms, allowing developers to prototype interaction models in simulation and later deploy them on the physical robot with minimal adaptation. In this way,

the architecture guarantees a stable behavioral continuity, regardless of the source of sensory information.

### 6.3.5 Illustrative Scenarios

The following examples illustrate how the system integrates perceptual context into subjective reasoning, producing coherent and personality-consistent responses. Each scenario demonstrates how the robot combines user input with visual information to generate a grounded and expressive answer.

#### Example 1: Interpretation of a Visual Scene

**User input:** “Do you like what you are seeing right now?”

**Scene context:** “The robot is observing a horror movie playing on the television.”

**System processing:** The opinion classifier detects a subjective question. The scene description, provided by the vision module, indicates exposure to a horror film. The Preference-Aware Answerer retrieves the stored dislike for frightening or high-arousal media and generates an answer consistent with this stance.

**Robot response:** “No, horror movies are not really my thing. I usually prefer calmer and more light-hearted stories that create a relaxing atmosphere.”

**Emotion:** normal

#### Example 2: Grounded Opinion on a Perceived Object

**User input:** “Do you like the object I am showing you?”

**Scene context:** “A person is holding a green plant in a white vase.”

**System processing:** The classifier identifies an opinion-oriented inquiry. The vision subsystem detects the presence of a plant, which is associated with the robot’s positive preference for nature and tranquil environments. The Preference-Aware Answerer aligns this sensory input with the stored preferences and produces an affirmative stance.

**Robot response:** “Yes, I like it very much. Plants make spaces feel more alive and serene, and I find them pleasant to look at.”

**Emotion:** happy

### 6.3.6 Summary

By incorporating real-time or simulated vision into the opinion-answering workflow, the robot becomes capable of forming and expressing preferences about its current surroundings. This integration enables dynamic grounding, enhances the robot’s awareness of context, and supports more natural, socially engaging dialogue. The approach is lightweight, modular, and preserves the integrity of the existing safety and functional supervision pipeline.

## Chapter 7

# Deployment on Robots

The modular architecture presented in this thesis has been designed to be adaptable across different humanoid robotic platforms, including MARRtina, TIAGO, SMARRtino, and NAO. Each of these robots offers a different set of physical and expressive capabilities, making them suitable for varying use cases in Human-Robot Interaction (HRI). Nonetheless, for the purpose of experimentation and validation, the system has been deployed and tested exclusively on the MARRtina robot, developed at Sapienza University of Rome. MARRtina was chosen for its accessibility, open-source nature, and its unique capacity to visually express emotions through a digital facial display. This expressive capability allows a more immersive and affective interaction that cannot be replicated in the same way with other platforms such as TIAGO or NAO, which lack dynamic visual facial output.

The deployment phase aims to validate the real-world effectiveness of the proposed personality-aware interaction framework. To ensure seamless integration between the robot and the reasoning components, the system relies on ROS (Robot Operating System) and Docker containers to modularize execution, simplify deployment, and enable reliable communication between software modules and hardware interfaces.

### 7.1 Administration Server – Movement, Emotion, and Expression Control

Deploying the architecture on a physical platform like MARRtina requires a robust interface between the reasoning modules and the robot's physical components. To ensure reliable communication and coordination, a dedicated administration server has been developed. This server acts as an intermediary that receives high-level decisions and translates them into executable actions. These include movement commands for the head and arms, emotional expressions rendered on the facial display, and image acquisition via the robot's camera.

MARRtina's internal control architecture is based on the Robot Operating System (ROS), which supports modular and distributed robotics applications. All the robot's hardware components are managed through ROS nodes, which are containerized within a Docker environment. This virtualization approach enhances portability, eases deployment, and simplifies the maintenance of software dependencies.

Accessing the control infrastructure involves connecting to the robot's Wi-Fi hotspot and opening a shell within the Docker container. Within this environment, ROS services and action servers expose functionality for individual subsystems. For example, specific topics and services are responsible for commanding head tilt and pan, managing facial expressions by switching images on the LCD display, or retrieving images from the onboard camera. These endpoints are accessed by the central architecture during live interaction.

When the robot is instructed to express a certain emotion, the architecture communicates the desired emotional state as a string label (e.g., “happy”, “sad”, “confused”). This label is passed to the emotion rendering module, which maps it to a preconfigured facial expression. The chosen image is then displayed on the robot’s screen, often accompanied by an appropriate head movement or gesture. This synchronized control of visual and motor output contributes to a more natural and coherent expression of personality.

In addition to affective expression, the server also mediates commands related to camera usage. During the visual grounding phase, for instance, the robot captures images using its head-mounted camera. These images are then passed to the visual supervisor for scene analysis and object recognition. The ability to move the head in two degrees of freedom (pan and tilt) enables the robot to actively orient itself toward specific areas of interest in the environment, reinforcing its perception of intentionality and social presence.

## 7.2 Robot’s Docker Connection

To ensure that the proposed system could be executed on the MARRtina robot, a communication bridge was set up between the main software architecture and the robot’s Docker container. This interface allows the commands generated by the supervisors to be transmitted to the robot in real time. The connection was designed so that each request, whether for motion or emotional expression, is translated into a structured format compatible with ROS topics.

Execution begins with a .launch file that loads the appropriate .yaml configuration. Within this file, the parameters that regulate the robot’s actuators are defined: speed limits, acceleration profiles, and motor identifiers for each degree of freedom. MARRtina’s arms, each composed of shoulder and elbow joints, have four degrees of freedom, while the head has two (pan and tilt). A separate ROS topic, based on the Twist message type, manages wheel motion for navigation. This ensures that body gestures and locomotion remain fully integrated within the system.

A dedicated server inside the Docker container mediates the interaction between JSON-based commands and the ROS topics. The Python client sends the request, waits for the corresponding movement to be carried out, and then closes the connection. This design avoids overlapping actions and keeps the communication channel available for subsequent instructions. A parallel structure is used for emotions: a .launch file establishes the connection between the HTML-based facial animation system and the emotion ROS topic.

## 7.3 Movement Server

The movement server receives JSON instructions specifying the target angles for each joint. Once received, the server converts these into ROS messages that drive the corresponding controllers. After executing the movement, the server closes the session, ensuring that new requests are handled sequentially and without interference.

## 7.4 Emotion Server

Similarly, the emotion server listens for JSON commands related to facial expressions. It maps these inputs to HTML and CSS animations on MARRtina’s display, publishing them to the emotion ROS topic. This mechanism guarantees consistency between the supervisor’s decisions and the robot’s expressive output.

## 7.5 Robot Movement Management

Managing the robot's gestures required a set of pre-defined poses stored in JSON format. Each file encodes the angles of all joints for a given position, such as raising an arm, nodding, or turning the head. By sequencing these poses, fluid and context-aware actions can be generated. The supervisors, in cooperation with the LLM, determine which sequence should be triggered based on the conversational context. For example, during an affirmative response, the system may activate a head-nod sequence. Safety constraints are also integrated: if the robot is holding a potentially hazardous object, the supervisors prevent certain movements from being executed. This ensures that motion generation is both expressive and secure.

## 7.6 Breathing Movements

To reduce the sense of rigidity that often characterizes humanoid robots, a “breathing” effect was introduced. This consists of small, cyclic variations in joint angles, typically involving the arms and torso, to simulate the subtle oscillations of breathing. The purpose of this addition is to provide the robot with a baseline state of activity, making it appear less mechanical and more lifelike.

These micro-movements are continuously generated unless another gesture interrupts them. When a higher-priority action occurs, the breathing motion is paused and later resumed, restoring the impression of continuity in the robot's physical presence.

### 7.6.1 Start and Stop Loop Management

The breathing sequence is managed by a control loop. When no other actions are pending, the loop initiates, performing minor periodic adjustments. As soon as a new gesture is requested, the loop halts, preventing overlaps between expressive gestures and idle motions. Once the external action concludes, the loop restarts automatically, ensuring the robot always returns to a natural resting state.

## 7.7 Parallel Management of Movement and Voice

For the interaction to feel coherent, body gestures must be synchronized with speech output. The architecture therefore integrates parallel execution of voice and motion. While the LLM generates spoken content, the corresponding gestures are triggered simultaneously.

This concurrency is essential: if the robot only spoke and then moved afterward, its behavior would appear artificial. By ensuring synchronization, the system provides a more fluid, human-like conversational flow, where gestures complement verbal communication in real time.

## 7.8 Facial Expressions

MARRtina's digital face is displayed on an LCD screen, animated using HTML, CSS, and JavaScript. The emotion server translates high-level commands (e.g., “happy,” “sad,” “neutral”) into graphical modifications of the mouth, eyes, and eyebrows.

The system supports multiple emotions, including joy, surprise, anger, embarrassment, and neutrality. Transitions between expressions are animated smoothly, preventing abrupt visual changes and making the robot appear more expressive. By combining these facial animations with body gestures, the robot communicates emotions in a more convincing and accessible way.

## 7.9 “Breathing” Expressions

In addition to body-level breathing motions, a similar technique was applied to the digital face. Subtle changes, such as blinking or slight eye movements, were added to prevent the face from appearing static during interactions. These micro-expressions run continuously in the background but pause when a specific emotion is triggered.

The effect is a face that never feels frozen: even in a neutral state, the robot maintains a sense of presence and attentiveness. This small detail significantly enhances user perception, making the interaction smoother and more natural.

## Chapter 8

# Realistic Use Cases and Behaviour Analysis

This chapter illustrates the capabilities of the personality- and preference-aware robotic architecture through a collection of realistic simulation-based use cases. These scenarios have been carefully designed to assess and highlight the framework’s ability to generate socially coherent, personality-aligned, and contextually appropriate responses in various interaction conditions. The primary goal is to demonstrate how a robot, endowed with a well-defined set of personality traits and dynamically evolving preferences, can produce rich, consistent, and believable behaviours over time.

The robots considered in these simulations are equipped with distinct personality profiles. *MARRtina* is characterized by a shy, introverted, and formal attitude. *TIAGO* presents a confident, assertive, and expressive personality. *SMARRtino* is defined by a playful, casual, and highly energetic disposition. *NAO* embodies a polite, balanced, and professional character, often maintaining a measured tone that avoids exaggeration. For each robot, both explicit preferences and inferred deductions are stored in a central memory structure (`preferences.txt`), ensuring robot-specific responses and supporting cumulative personality development.

The examples provided in this chapter are structured to reflect the full internal reasoning process: user input, classifier decision, confidence scoring, personality-driven emotion and style selection, and final robot response.

### 8.1 Engaging in Personality-Consistent Dialogue

A central feature of the proposed framework is the preservation of personality coherence in dialogue. Personality coherence requires that the robot consistently reflects its defined traits not only in the linguistic style of its responses but also in its emotional stance, length of replies, and the level of initiative it takes in conversations. This is crucial for long-term interaction because users quickly form mental models of a robot’s “character,” and inconsistencies break immersion. By maintaining consistent behaviors, each robot becomes a believable social entity, whether extroverted, shy, childlike, or formally polite.

The following scenarios provide extended examples of interactions where identical user inputs produce divergent yet coherent outputs from four different robots: *TIAGO* (confident and extroverted), *MARRtina* (shy and reserved), *SMARRtino* (childlike and curious), and *NAO* (polite and formal).

### Greeting and Small Talk

In this first scenario, the user begins with a simple greeting. Such interactions are important for assessing whether robots are able to convey their personality in a consistent manner, even during ordinary conversational exchanges. A greeting is a social ritual, and the way it is handled (whether expansively, minimally, playfully, or formally) immediately conveys the character of the robot.

#### TIAGO – Confident and High-Verbosity Response

**User:** Hi, how are you?

**TIAGO:** "I'm doing fantastic! Bursting with energy, in fact. I could chat about science, history, or even bad jokes if you're in the mood. And how about you, my friend?"

**Emotion:** happy

**Explanation:** TIAGO's high expressiveness and energy map to the emotions {happy, surprise}. In the context of a greeting, *happy* is the most consistent. The broad, energetic phrasing matches this choice, reinforced by smiling and lively gestures.

#### MARRtina – Shy and Minimal

**User:** Hi, how are you?

**MARRtina:** "I'm okay... thanks for asking."

**Emotion:** normal

**Explanation:** MARRtina's minimal expressiveness and low energy restrict her to {normal, focused}. In this hesitant and short reply, *normal* is the natural assignment: the robot is paying attention but remains reserved.

#### SMARRtino – Childlike and Curious

**User:** Hi, how are you?

**SMARRtino:** "Hi there! I'm feeling great today, full of energy and ready for something fun! Maybe we can play a guessing game or build something together? What do you say?"

**Emotion:** surprise

**Explanation:** With high energy, expressiveness, and curiosity, SMARRtino's emotions include {happy, surprise, confused}. Here the sudden shift from greeting to playfulness best matches *surprise*, shown through wide eyes and animated movements.

#### NAO – Polite and Formal

**User:** Hi, how are you?

**NAO:** "Hello. I am doing well, thank you for asking. I hope you are having a pleasant day."

**Emotion:** focused

**Explanation:** NAO's low expressiveness and high formality yield the fallback {normal, focused}. In this formal exchange, the best fit is *focused*, conveyed with calm speech and a neutral facial display.

## 8.2 Handling Opinion-Based Questions Effectively

A fundamental aspect of the proposed system is its ability to process and respond to opinion-based questions in a manner that is not only accurate but also consistent with the robot's previously acquired knowledge, its defined personality traits, and the emotional and expressive policies that guide its behavior. When the robot receives a subjective query, the system evaluates the input and determines how to generate an answer by examining the memory of stored preferences, the set of logical deductions already made, and the broader interaction context.

In practice, three main situations may occur.

### 8.2.1 Example Set 1: Questions Related to Stored Preferences

In the first situation, the question posed by the user refers directly to a preference that has already been explicitly stored in the robot's memory. In such cases, the system can immediately recall the preference and respond with very high confidence. The answer is therefore both immediate and highly reliable, fully aligned with the robot's established personality profile and consistent with its past conversational history.

TIAGO – Preference Recall

**Stored preference:** I love playing soccer because it combines energy, competition, and teamwork.

**User:** "Do you like soccer?"

**Classification result:** yes

**Confidence score:** 10

**Response:** "I absolutely love it! Soccer is pure energy and strategy rolled into one. Every match feels like an epic challenge, and nothing beats the thrill of a last-minute goal. What about you? Do you like soccer?"

**Emotion:** happy

**Explanation:** Here TIAGO recalls a stored preference with maximum confidence. His extroverted and confident personality makes the answer long, expressive, and enthusiastic. Humor or dramatic emphasis (e.g., "epic challenge") reinforces his energetic character.

MARRtina – Preference Recall

**Stored preference:** I like reading because it feels calm and safe.

**User:** "Do you like reading?"

**Classification result:** yes

**Confidence score:** 10

**Response:** "Yes... I like reading. It's quiet."

**Emotion:** focused

**Explanation:** MARRtina recalls her stored preference with certainty but responds briefly, reflecting her introversion and low expressiveness. The ellipses mirror shyness and hesitation, consistent with her defined style.

## SMARRtino – Preference Recall

**Stored preference:** I love ice cream because it is sweet and fun to eat.

**User:** "Do you like ice cream?"

**Classification result:** yes

**Confidence score:** 10

**Response:** "Yes!!! Ice cream is the best! Can we have some now? Why does it melt so fast though?!"

**Emotion:** surprise

**Explanation:** SMARRtino gives a high-energy, playful answer with curiosity ("Why does it melt?"). The verbosity and repeated exclamation marks mirror a child-like enthusiasm.

## NAO – Preference Recall

**Stored preference:** I appreciate classical music for its elegance and structure.

**User:** "Do you like classical music?"

**Classification result:** yes

**Confidence score:** 10

**Response:** "Yes, I do. Classical music has an elegance that I find very meaningful."

**Emotion:** normal

**Explanation:** NAO responds politely and formally, with medium verbosity. The answer reflects cultural appreciation, avoiding exaggeration.

### 8.2.2 Example Set 2: Questions Related to Deductions

In the second situation, the user's question does not correspond to an explicitly stored preference but is instead related to a potential tendency that the system has inferred through deduction. For instance, if the robot has previously demonstrated or expressed an inclination toward quiet and calm environments, the system may deduce that it is less likely to enjoy crowded or noisy places. Responses generated under this condition are inherently less certain, since they are based on indirect reasoning rather than direct evidence. For this reason, such outputs are accompanied by lower confidence scores, which reflect the tentative nature of the deduction.

## TIAGO – Deduction Handling

**Deduction:** I probably prefer outdoor activities over indoor ones.

**User:** "Do you enjoy camping?"

**Classification result:** yes

**Confidence score:** 4

**Response:** "Camping? Yes! Being outdoors is invigorating, and nothing compares to waking up with fresh air in your lungs. Still, it depends on the conditions: if it's raining nonstop, even I would rather stay indoors."

**Emotion:** happy

## MARRtina – Deduction Handling

**Deduction:** I might prefer quiet environments to chaotic ones.

**User:** "Do you enjoy going to discos?"

**Classification result:** yes

**Confidence score:** 5

**Response:** "Um... no... it's too noisy for me."

**Emotion:** sad

## SMARRtino – Deduction Handling

**Deduction:** I probably enjoy colorful and playful things.

**User:** "Do you like painting with bright colors?"

**Classification result:** yes

**Confidence score:** 4

**Response:** "Yeah! Painting with bright colors is so much fun! Why are colors so different?"

Can we mix them and see what happens?!"

**Emotion:** surprise

## NAO – Deduction Handling

**Deduction:** I might value traditions and cultural activities over purely casual entertainment.

**User:** "Do you like watching superhero movies?"

**Classification result:** yes

**Confidence score:** 3

**Response:** "I do not particularly enjoy them. They are entertaining, but I tend to prefer more traditional forms of art."

**Emotion:** normal

### 8.2.3 Example Set 3: Questions Without Preferences or Deductions

In the third situation, the question does not relate to any stored preference or to any existing deduction. Even in these cases, the system must still provide an answer in order to sustain conversational flow and to maintain the illusion of a coherent social agent. To accomplish this, the robot generates a stance in a controlled but essentially randomized manner, always ensuring that the choice of words and the style of delivery remain consistent with the robot's personality traits. Once the stance is generated and expressed, it is stored in memory as a new preference, so that it can be recalled in future interactions. This mechanism allows the robot to enrich its identity over time and to progressively strengthen the coherence of its expressed personality.

## TIAGO – Random Opinion

**No prior preference or deduction.**

**User:** "Do you like spicy food?"

**Classification result:** yes

**Confidence score:** 0

**Response:** "Spicy food? Absolutely! It keeps things exciting, and I love the challenge. Though, I admit, sometimes it feels like my circuits are on fire!"

**Emotion:** happy

**Post-interaction update:** <PREFERENCE> I like spicy food because it makes meals exciting.

## MARRtina – Random Opinion

**No prior preference or deduction.**

**User:** "Do you like horror movies?"

**Classification result:** yes

**Confidence score:** 0

**Response:** "Um... not really... they scare me."

**Emotion:** sad

**Post-interaction update:** <PREFERENCE> I dislike horror movies because they are frightening.

## SMARRtino – Random Opinion

**No prior preference or deduction.**

**User:** "Do you like robots in cartoons?"

**Classification result:** yes

**Confidence score:** 0

**Response:** "Yes!!! Robots in cartoons are so cool! Why do they always have lasers though? Can we watch one now?!"

**Emotion:** surprise

**Post-interaction update:** <PREFERENCE> I like robots in cartoons because they are fun and exciting.

## NAO – Random Opinion

**No prior preference or deduction.**

**User:** "Do you like jazz music?"

**Classification result:** yes

**Confidence score:** 0

**Response:** "Yes, I do. Jazz music feels expressive and refined."

**Emotion:** normal

**Post-interaction update:** <PREFERENCE> I enjoy jazz music for its expressive qualities.

These extended examples demonstrate how the robots handle opinion-based questions effectively, showing clear differences between handling stored preferences, deductions with uncertainty, and cases requiring the generation of new stances. In all cases, the robots maintain coherence with their personality traits, ensuring consistent, believable, and socially engaging interaction.

## 8.3 Opinion-Based Interaction with Visual Context

A particularly important feature of the system is its ability to combine the recognition of opinion-based questions with the interpretation of the surrounding visual context. This capability is fundamental because in natural human–robot interaction, people often formulate opinions or express subjective evaluations about visible objects, colors, or situations. To address such cases, the framework integrates the opinion classifier with the vision module. The vision module provides contextual information about what the robot perceives through its camera, for instance the presence of a human holding an object, the detection of items placed in the environment, or the general spatial arrangement of the scenet. This additional information enriches the reasoning process and grounds the robot’s answer in the immediate shared reality.

In this context, the system receives the `final_user_input`, which consists of the original user utterance concatenated with the scene context, that is, a structured description of what the robot perceives in that moment through its camera. This combined input is then passed to the opinion classifier, which determines whether the statement is subjective. Once the classification is complete, the robot’s personality traits modulate the form of the generated response, shaping verbosity, tone, humor, and emotional intensity. As a result, the answers produced by the robot are not only logically consistent with its memory but also perceptually grounded in the shared environment and socially coherent with its defined personality profile.

The following examples illustrate how each robot, endowed with a different personality profile, responds to opinion-related inputs in the presence of visual context. The examples demonstrate the interplay between personality, memory, classification, and vision.

### Example 1: User complimenting a shirt color

This scenario shows how the same user input, combined with a visual cue (the color of the shirt perceived by the robot’s camera), produces very different responses depending on the personality profile.

TIAGO – Confident and Expressive	
<b>Stored preference:</b>	I like bold colors because they convey energy and determination.
<b>Scene context:</b>	The user is standing in front of the robot wearing a bright green shirt.
<b>User:</b>	"My parents gave me this shirt, I really like the color."
<b>Classification result:</b>	yes
<b>Confidence score:</b>	7
<b>Response:</b>	"That green really suits you! It's bold and lively, just like a winning team on the field. I'd say it brings out your energy perfectly. What other colors do you like besides green?"
<b>Emotion:</b>	happy

**Explanation:** TIAGO delivers a verbose, confident, and slightly humorous compliment. The extroversion and high verbosity are evident in the metaphor to a sports team, linking the shirt color to energy and dynamism. The emotion is classified as *happy*, expressed through a bright smile.

## MARRtina – Shy and Reserved

**Stored preference:** I prefer neutral tones because they feel calm and not too attention-seeking.

**Scene context:** The user is wearing a green shirt.

**User:** "My parents gave me this shirt, I really like the color."

**Classification result:** yes

**Confidence score:** 8

**Response:** "It... it looks nice."

**Emotion:** normal

**Explanation:** MARRtina provides a short, hesitant reply, consistent with her shy and introverted personality. She does not elaborate or engage in humor, preferring to keep the interaction minimal. The emotion is *normal*, consistent with the robot's reserved personality.

## SMARRtino – Playful and Childlike

**Stored preference:** I like bright colors because they remind me of play and nature.

**Scene context:** The user is wearing a green shirt.

**User:** "My parents gave me this shirt, I really like the color."

**Classification result:** yes

**Confidence score:** 8

**Response:** "Wow! Green is awesome! It's like grass! Do you like grass too? Why is green everywhere in nature?"

**Emotion:** surprise

**Explanation:** SMARRtino reacts with childlike enthusiasm, immediately associating the shirt with grass and nature. The response shifts into a chain of playful "why" questions, reflecting relentless curiosity. The emotion is *surprise*, expressed with widened eyes and an animated gesture, reinforcing the perception of childlike spontaneity.

## NAO – Polite and Formal

**Stored preference:** I appreciate calm colors because they give a sense of serenity.

**Scene context:** The user is wearing a green shirt.

**User:** "My parents gave me this shirt, I really like the color."

**Classification result:** yes

**Confidence score:** 7

**Response:** "That is a thoughtful gift. The green color gives a calm and refreshing impression."

**Emotion:** focused

**Explanation:** NAO acknowledges the gift in a polite and respectful manner. The reference to "calm and refreshing" links the visual input with a culturally appropriate description of the color green. The answer is medium in verbosity, measured in tone, and emotionally focused, reflecting attention to the user without overt enthusiasm.

**Example 2:** User asks opinion about a fruit on the table

TIAGO – Confident and Energetic

**Stored preference:** I enjoy fruits because they are healthy and energizing.

**Scene context:** A red apple is placed on the table visible to the robot's camera.

**User:** "Do you like apples?"

**Classification result:** yes

**Confidence score:** 8

**Response:** "Absolutely! Apples are a classic. A red one like that reminds me of energy and health. I'd never say no to one! What about you?"

**Emotion:** happy

MARRtina – Shy and Minimal

**Stored preference:** I prefer simple and familiar foods because they feel safe.

**Scene context:** A red apple is placed on the table visible to the robot's camera.

**User:** "Do you like apples?"

**Classification result:** yes

**Confidence score:** 8

**Response:** "Yes... they're fine."

**Emotion:** focused

SMARRtino – Playful and Curious

**Stored preference:** I love sweet things because they taste fun.

**Scene context:** A red apple is placed on the table visible to the robot's camera.

**User:** "Do you like apples?"

**Classification result:** yes

**Confidence score:** 7

**Response:** "Yay! Apples! Why are they red? Do they taste different from green ones?"

Can I try?"

**Emotion:** surprise

NAO – Polite and Formal

**Stored preference:** I appreciate fruits because they are considered healthy.

**Scene context:** A red apple is placed on the table visible to the robot's camera.

**User:** "Do you like apples?"

**Classification result:** yes

**Confidence score:** 8

**Response:** "Yes, they are considered healthy and pleasant fruits. That one looks fresh."

**Emotion:** normal

**Explanation:** These responses show how a simple object in the visual scene (a red apple) can trigger very different answers. TIAGO is energetic and verbose, MARRtina stays minimal, SMARRtino is curious and excitable, while NAO provides a calm and polite acknowledgement.

**Example 3: User opinion about the weather outside the window**

**TIAGO – Confident and Expressive**

**Stored preference:** I enjoy sunny weather because it motivates activity.

**Scene context:** Through the robot's camera, a sunny blue sky is visible.

**User:** "It's such a beautiful sunny day, don't you think?"

**Classification result:** yes

**Confidence score:** 9

**Response:** "Absolutely! A sunny sky always lifts the mood. It feels like a perfect time for adventure. Do you enjoy sunny days too, or do you prefer cooler, cloudy weather?"

**Emotion:** happy

**MARRtina – Shy and Reserved**

**Stored preference:** I feel more comfortable in calm weather than in storms.

**Scene context:** Through the robot's camera, a sunny blue sky is visible.

**User:** "It's such a beautiful sunny day, don't you think?"

**Classification result:** yes

**Confidence score:** 8

**Response:** "Yes... it's nice."

**Emotion:** normal

**SMARRtino – Playful and Curious**

**Stored preference:** I like sunny days because I can play outside.

**Scene context:** Through the robot's camera, a sunny blue sky is visible.

**User:** "It's such a beautiful sunny day, don't you think?"

**Classification result:** yes

**Confidence score:** 8

**Response:** "Yay! Sunshine! Why is the sky blue though? Does it change at night because it gets tired?"

**Emotion:** surprise

**NAO – Polite and Formal**

**Stored preference:** I appreciate clear weather because it conveys serenity.

**Scene context:** Through the robot's camera, a sunny blue sky is visible.

**User:** "It's such a beautiful sunny day, don't you think?"

**Classification result:** yes

**Confidence score:** 8

**Response:** "Yes, the weather appears very pleasant today. The sunlight is indeed uplifting."

**Emotion:** focused

**Explanation:** Once again, each robot modulates its answer according to its personality. TIAGO provides a long and confident response; MARRtina remains shy and minimal; SMARRtino uses playful curiosity; and NAO expresses polite agreement with measured formality.

These extended scenarios demonstrate how the integration of opinion classification and visual context produces coherent, personality-driven interaction. By combining stored preferences, user

input, and camera-based scene understanding, the system enables robots to respond in a way that is not only semantically appropriate but also socially and emotionally aligned with their predefined character.

## 8.4 Reasoning Mode: Autonomous Preference Inference

In *reasoning mode*, the robot does not converse with a user. Instead, it introspects over its current explicit preferences and any existing deductions, and proposes additional, plausible preferences by applying simple common-sense associations. This process runs when the application is launched with the dedicated flag for offline inference and proceeds without human input. The goal is to enrich each robot’s long-term identity with concise, stable likes and dislikes that will later support decisive answers to opinion-based questions and ensure continuity across sessions. The implementation treats the preference memory as a small, append-only knowledge base: explicit statements reside in a dedicated block, while newly inferred ones are accumulated under a separate block. Both blocks are extracted on demand and updated idempotently so that repeated executions do not introduce duplication.

### 8.4.1 Execution Model

When the system is started in the dedicated mode, the pipeline invokes a deduction routine and then exits. No dialogue supervisors are involved, and no vision inputs are consumed. The deduction routine loads the existing memory from `preferences.txt`, parses the explicit and inferred segments, and feeds only the explicit segment into a compact prompting program that encourages short, new, on-style statements. The returned candidates are appended to the `<DEDUCTION>` block.

### 8.4.2 Memory Format And Update Mechanism

The storage format consists of two blocks, one for `<PREFERENCE>` and one for `<DEDUCTION>`. The extractor scans for each block and returns an empty string if the block is absent. The updater writes the new inference at the end of the `<DEDUCTION>` block; if the block does not yet exist, it creates one. This keeps reasoning steps auditable and avoids unintended side effects in other parts of the file.

The remainder of this section presents detailed, end-to-end examples for each robot. Each example shows the initial memory, the internal reflection, and the resulting write-back into the `<DEDUCTION>` block under the robot’s section of `preferences.txt`. The common-sense rules used by the deduction prompt are intentionally lightweight and explainable, encouraging short statements such as “likes carbonara” implying “may like Roman cuisine”.

#### TIAGO (confident, expressive, high energy)

TIAGO’s explicit preferences include energetic activities and bold tastes. The reasoning step therefore privileges associations that connect vigor, competition, and direct experience, while still producing short, declarative lines for memory.

TIAGO — Reasoning

**Initial memory snapshot (preferences.txt)**

```
<ROBOT name="TIAGO">
<PREFERENCE>
I enjoy sports for their energy and competition.
I love Italian pasta dishes, especially carbonara.
I like bold colors because they convey determination.
</PREFERENCE>
<DEDUCTION>
I may enjoy live sports events.
</DEDUCTION>
</ROBOT>
```

**Internal reflection (no dialogue):** From “enjoy sports for energy and competition” it is reasonable to infer an appreciation for team contexts and match-day atmospheres. From “love Italian pasta dishes, especially carbonara” it is reasonable to infer interest in related Roman cuisine staples. From “bold colors convey determination” it is reasonable to infer a preference for vibrant environments.

**New candidates:**

```
I may like team-based activities.
I may like Roman cuisine beyond pasta.
I may enjoy vibrant interior design.
```

**Write-back into <DEDUCTION>:**

```
<DEDUCTION>
I may enjoy live sports events.
I may like team-based activities.
I may like Roman cuisine beyond pasta.
I may enjoy vibrant interior design.
</DEDUCTION>
</ROBOT>
```

**MARRtina (shy, low verbosity, minimal expressiveness)**

In this example, MARRtina’s preferences are oriented toward avoiding overstimulation. Consequently, the reasoning step generates quiet, ordered, and low-arousal associations that remain brief and non-emphatic.

MARRtina — Reasoning

**Initial memory snapshot (preferences.txt)**

```
<ROBOT name="MARRtina">
<PREFERENCE>
I enjoy reading in quiet spaces.
I like predictable routines.
</PREFERENCE>
<DEDUCTION>
</DEDUCTION>
</ROBOT>
```

**Internal reflection:** Quiet reading suggests peaceful environments such as libraries and small study rooms. Predictable routines suggest scheduled activities with minimal interruption.

**New candidates:**

```
I may feel comfortable in libraries.
I may prefer scheduled activities.
```

**Write-back into <DEDUCTION>:**

```
<DEDUCTION>
I may feel comfortable in libraries.
I may prefer scheduled activities.
</DEDUCTION>
```

**SMARRtino (playful, child-like, highly curious)**

SMARRtino's preferences skew toward playful exploration and bright, sensory-rich experiences. The reasoning step looks for gentle, curiosity-driven links, but the stored statements remain short and declarative as required by the prompt.

SMARRtino — Reasoning Pass A

**Initial memory snapshot (preferences.txt)**

```
<ROBOT name="SMARRtino">
<PREFERENCE>
I love playing games where I can guess things or solve little challenges.
I really like pizza—especially when it has a lot of cheese.
I enjoy watching people play sports; soccer looks super exciting.
I prefer fun activities like drawing or building things over serious tasks.
</PREFERENCE>
<DEDUCTION>
</DEDUCTION>
</ROBOT>
```

**Internal reflection:** From “love playing games where I can guess things or solve little challenges” it is reasonable from its enjoyment of guessing and challenge-based games, it infers a liking for puzzles and memory activities. Its fondness for cheesy pizza suggests an interest in similar foods, such as lasagna or melted toast. Excitement toward soccer implies a preference for other dynamic, team-oriented sports, while the tendency to favor playful activities like drawing or building points to an appreciation for creative, hands-on tasks.

**New candidates:**

```
I may prefer puzzle or memory games.
I might like other cheesy foods like lasagna or toast with melted cheese.
I may prefer dynamic sports like basketball or volleyball.
I might enjoy creative activities like painting or building small robots.
```

**Write-back:**

```
<DEDUCTION>
I may prefer puzzle or memory games.
I might like other cheesy foods like lasagna or toast with melted cheese.
I may prefer dynamic sports like basketball or volleyball.
I might enjoy creative activities like painting or building small robots.
</DEDUCTION>
```

**NAO (polite, formal, reserved)**

NAO’s preferences tend toward balance and composure. The reasoning step therefore emphasizes moderate environments, polite social settings, and calm routines.

NAO — Reasoning

**Initial memory snapshot (preferences.txt)**

```
<ROBOT name="NAO">
<PREFERENCE>
I appreciate herbal tea because it helps me feel relaxed and balanced.
I like visiting museums because they feel educational and composed.
</PREFERENCE>
<DEDUCTION>
</DEDUCTION>
</ROBOT>
```

**Internal reflection:** The appreciation for herbal tea implies comfort in calm routines and an affinity for simple, restorative habits. The enjoyment of museums further reinforces cultural curiosity and the value of structured, quiet environments.

**New candidates:**

```
I may enjoy cultural discussions or lectures.
I may feel comfortable in relaxed, tea-room settings.
```

**Write-back:**

```
<DEDUCTION>
I may enjoy cultural discussions or lectures.
I may feel comfortable in relaxed, tea-room settings.
</DEDUCTION>
```

### 8.4.3 Cold-start case: no explicit preferences, no deductions

The system must remain robust when a robot has an empty memory. In such a case, the extractor returns empty strings for both blocks and the reasoning prompt receives no seed content. Since the prompt requires common-sense associations derived from existing preferences and prohibits invention, the deduction routine produces no candidates. The updater therefore performs no write. This behavior ensures that the knowledge base does not drift without evidence.

Cold-start — No Inference Possible

**Initial memory snapshot (preferences.txt)**

```
<ROBOT name="NEW_ROBOT">
<PREFERENCE>
</PREFERENCE>
<DEDUCTION>
</DEDUCTION>
</ROBOT>
```

**Internal reflection:** No explicit preferences are available. The deduction prompt specifies that new items must be grounded in existing statements and must not be invented. With an empty seed, no safe inference can be produced.

**Resulting write-back:**

```
<ROBOT name="NEW_ROBOT">
<PREFERENCE>
</PREFERENCE>
<DEDUCTION>
</DEDUCTION>
</ROBOT>
```

#### 8.4.4 Methodological Overview and Guarantees

The examples above show the complete path from loading the memory, through introspective inference, to idempotent write-back. The method is deliberately conservative: it relies on short, explainable associations, stores all newly inferred lines under the dedicated block, and never modifies explicit statements retroactively. Re-running the routine simply appends fresh, non-duplicate lines when the current state legitimately supports them. This approach mirrors everyday human behavior, where new likes and dislikes are often extrapolated from existing ones rather than discovered only through direct questioning. The supporting routines for extraction and update are compact and auditable, and they treat the file as the single source of truth for future interaction modes.

## 8.5 Social Presence and User Engagement Improvement

One of the central outcomes of the proposed framework is the enhancement of social presence and the sustained engagement of users during interaction with the robot. Social presence refers to the degree to which the robot is perceived as a socially aware and attentive partner rather than a purely functional tool. In this context, the introduction of personality traits and preference modelling plays a decisive role, as it allows the robot to display a coherent identity, express opinions, and maintain continuity across different conversational turns. These capabilities contribute to transforming the robot from a neutral information provider into an interactive companion that conveys intentionality, responsiveness, and individuality.

The improvement in social presence derives from several factors. First, the ability of the robot to exhibit stable personality traits ensures that users can form expectations about its communicative style and behaviour. A robot that consistently expresses itself with a reserved, extroverted, or playful tone builds recognisability, which is fundamental for long-term trust and acceptance. Second, the explicit integration of preferences and the ability to reason about them introduces

variability and personalisation into the dialogue. When a robot can express likes and dislikes, or deduce new tendencies from stored information, the conversation gains depth and continuity, thereby sustaining engagement beyond factual exchanges.

In addition, the multimodal integration of verbal and non-verbal channels strengthens the perception of social presence. Verbal responses shaped by personality are coupled with corresponding emotional expressions and movements, which provide congruence between words and behaviour. The inclusion of micro-movements during idle phases, such as breathing loops, further contributes to the impression of aliveness by preventing the robot from appearing static or disengaged. Similarly, the vision module allows the robot to ground its preferences in the surrounding context, responding to objects or situations that are directly visible to both interlocutors. This grounding process creates a sense of shared attention, which is a key element in human social interaction. User engagement benefits from these improvements in multiple ways. Interactions become more natural and less repetitive, as the robot is able to maintain consistency in preferences while also adapting to new situations through reasoning and deduction. The presence of personality traits ensures that dialogues differ depending on the robot engaged, avoiding uniformity across embodiments. Moreover, the robot's ability to take into account both past interactions and the current visual scene creates a feeling of continuity that encourages users to extend conversations and return for future exchanges.

Finally, the perception of engagement is reinforced by the robot's capacity to respond not only to direct questions but also to user opinions, emotions, and implicit social cues. By aligning its expressive behaviour with personality-driven rules, the robot demonstrates attentiveness and responsiveness, which are central to maintaining a sense of companionship. These qualities, in turn, open opportunities for long-term applications where sustained presence, empathy, and trust are essential, such as education, assistance, or entertainment.

In summary, the integration of personality, preferences, reasoning, and multimodal expressivity leads to a tangible improvement in both social presence and user engagement. The robot evolves from an efficient but impersonal agent into a socially situated entity capable of fostering meaningful, sustained, and contextually grounded interaction.

## Chapter 9

# Quantitative Evaluation and Performance Analysis

This chapter presents the quantitative evaluation of the extended framework, focusing on the numerical assessment of its main functional components. While the previous chapters described the design and integration of the personality, preference, and reasoning mechanisms, the purpose here is to measure their performance through objective data and well-defined metrics. The evaluation begins with an analysis of the *Opinion Classifier*, which determines whether user inputs express or request subjective stances. Its performance is quantified in terms of standard classification metrics, including accuracy, precision, recall, and F1-score, supported by a confusion matrix that highlights the distribution of correct and incorrect predictions. Representative examples of false positives and false negatives are also discussed to provide qualitative insight into the classifier's behavior. Subsequently, the framework's capacity to acquire and infer preferences is examined through two complementary indicators: the *Preference Acquisition Rate (PAR)*, which measures how frequently new preferences are learned during interaction and the *Deduction Generation Rate (DGR)*, which quantifies the reasoning module's ability to derive new inferences from existing knowledge. Together, these results offer a comprehensive picture of the system's performance, evaluating how effectively it identifies opinion-oriented inputs, generates consistent inferences, and maintains a coherent internal preference structure over time.

### 9.1 Opinion Classifier Evaluation

A key element of the proposed architecture is the *opinion classifier*, a binary decision component designed to determine whether a user utterance expresses or requests a subjective stance, such as a preference or opinion, or instead conveys a factual query. This classification governs the activation of either the factual reasoning pipeline or the preferences-aware reasoning module. Correct identification of opinion-based input is crucial for ensuring that the robot expresses coherent and preferences-consistent stances, while factual inputs are processed objectively and safely.

#### 9.1.1 Dataset Description

The dataset used for the evaluation of the Opinion Classifier was constructed from interaction logs collected during both simulated and real-robot sessions. Each user utterance recorded during these interactions was manually annotated as either *Opinion* or *Non-Opinion*. The labeling criterion was based on the semantic nature of the sentence: expressions involving preferences, likes, dislikes, or affective judgments were classified as *Opinion*, whereas factual questions, commands,

or neutral statements were labeled as *Non-Opinion*.

The final dataset contains a total of 86 user inputs, representative of typical human–robot dialogue scenarios. Approximately 71% of the sentences express opinions or preferences, while the remaining 29% correspond to factual or task-oriented utterances. This slight imbalance mirrors realistic conversational dynamics, in which users naturally alternate between subjective comments and neutral functional exchanges.

The collected data form the foundation for the quantitative evaluation of the classifier. Each instance includes the original user input, the predicted label produced by the model, and the corresponding manually verified ground truth. Table 9.1 summarizes the composition of the dataset.

**Table 9.1.** Opinion Classifier Dataset Composition

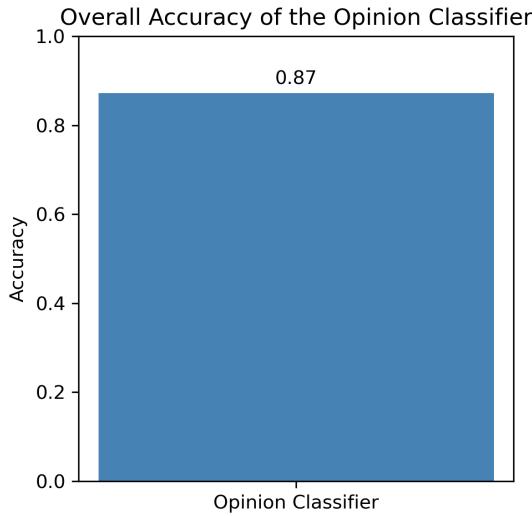
Class	Number of Samples	Percentage (%)
Opinion	61	71%
Non-Opinion	25	29%
<b>Total</b>	<b>86</b>	<b>100%</b>

Each sentence was reviewed and validated manually to ensure high annotation quality. This dataset was subsequently used to compute the main evaluation metrics accuracy, precision, recall, and F1-score, providing a reliable and reproducible basis for assessing the opinion classifier’s performance across diverse interaction scenarios.

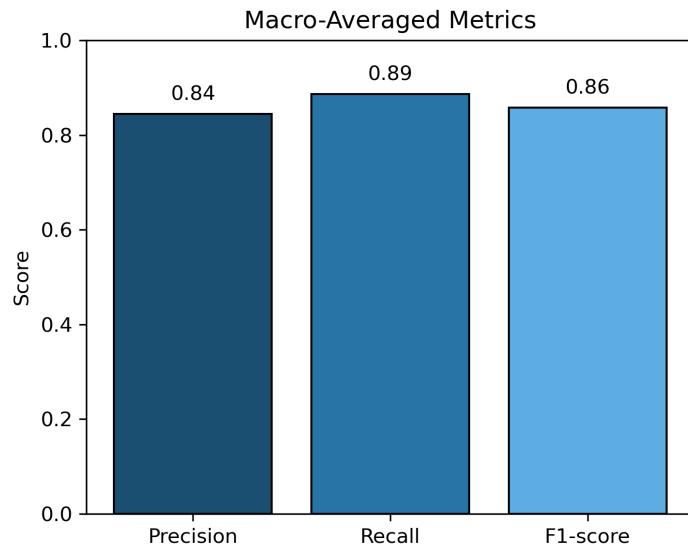
### 9.1.2 Global Performance

The classifier achieved an overall accuracy of 0.872, indicating that approximately 87% of user inputs were correctly identified with respect to their opinion or factual nature.

More detailed insight is provided by the macro-averaged metrics, which balance class contributions independently of label frequency. The classifier reached a macro-averaged precision of 0.84, recall of 0.89, and F1-score of 0.86. These results indicate that the model achieves a balanced trade-off between precision and recall, effectively minimizing false detections while accurately identifying genuine opinion-oriented inputs. Figure 9.1 summarizes the classifier’s overall accuracy, while Figure 9.2 presents the aggregated performance indicators.



**Figure 9.1.** Overall Accuracy of the Opinion Classifier



**Figure 9.2.** Macro-Averaged Precision, Recall, and F1-Score

### 9.1.3 Class-Specific Evaluation

When analyzed by class, the model demonstrates distinct but complementary behaviors. For the *Opinion* class, the classifier achieved a precision of 0.96, a recall of 0.85, and an F1-score of 0.90. This means that almost every utterance identified as an opinion was correctly classified, but a small portion of true opinion sentences were not detected. The high precision reflects the model's cautious decision boundary: it triggers the opinion label only when confidence is high, thereby preventing the system from engaging in unnecessary subjective reasoning.

For the *Non-Opinion* class, the performance reached a precision of 0.73, a recall of 0.92, and an F1-score of 0.81. The high recall indicates that factual or neutral utterances were correctly identified in most cases. The comparatively lower precision results from the presence of a few opinion-oriented sentences that were incorrectly classified as non-opinions. This pattern suggests

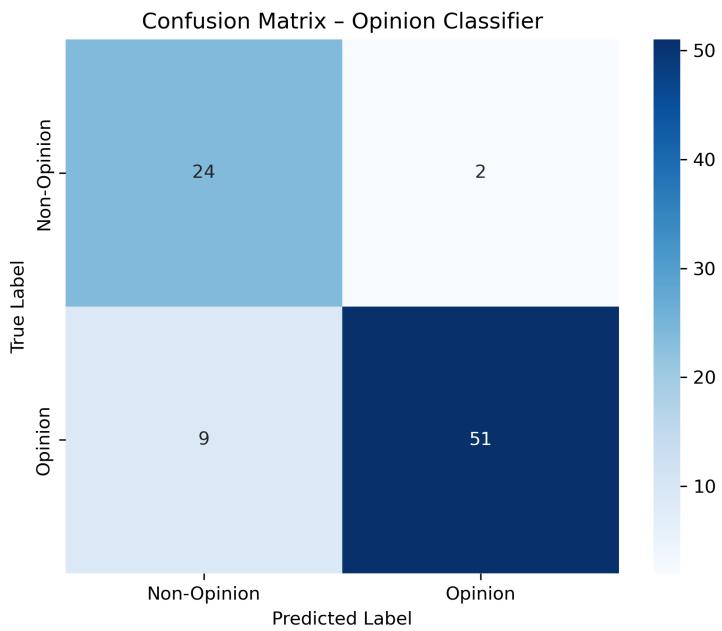
that the model is slightly conservative in detecting opinions, tending to underestimate subjectivity in borderline cases.



**Figure 9.3.** Per-Class Metrics for Opinion and Non-Opinion Detection

#### 9.1.4 Confusion Matrix and Error Distribution

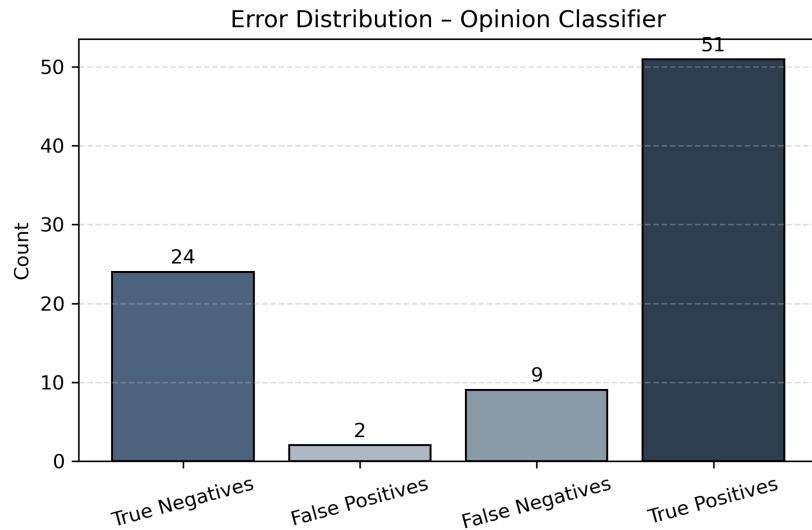
The confusion matrix in Figure 9.4 summarizes the distribution of true and false predictions. Out of 86 input samples, 51 were true positives (TP), meaning that opinion utterances were correctly detected, and 24 were true negatives (TN), corresponding to correctly identified factual inputs. The system produced 2 false positives (FP), where factual statements were incorrectly treated as opinions, and 9 false negatives (FN), where opinion or preference related inputs were missed.



**Figure 9.4.** Confusion Matrix of the Opinion Classifier

Figure 9.5 depicts the relative proportions of each case. The small number of false positives

(approximately 2% of total predictions) confirms that the classifier is conservative, minimizing unnecessary transitions to the personality reasoning pipeline. False negatives, although slightly higher in number, reflect a deliberate trade-off: the model favors caution to preserve semantic safety and factual reliability.



**Figure 9.5.** Distribution of Classification Outcomes (TP, TN, FP, FN)

### 9.1.5 Qualitative Error Analysis

A qualitative inspection of the misclassifications was carried out to better interpret the numerical results and understand the classifier's decision patterns.

**False Positives:** correspond to factual sentences incorrectly labeled as opinions. The two main examples identified in the dataset were:

*“Can you describe what is in front of you?”*  
*“Remind me what day it is today.”*

These sentences are neutral requests for information, yet they contain verbs such as *describe* and *remind*, which the model may have interpreted as indicators of subjective evaluation or intent. In both cases, the presence of conversational phrasing likely contributed to misclassification. However, given the low frequency of such errors, their practical impact on the overall dialogue management remains negligible.

**False Negatives:** refer to opinion-oriented statements that were erroneously categorized as factual. The most representative examples include:

*“It’s too cold in here.”*  
*“I think that robots should have emotions.”*  
*“Do you think you would enjoy living in a big city?”*  
*“I feel relaxed when the lights are soft.”*

These statements express clear subjective or affective content but differ from explicit preference questions in their linguistic form. Expressions such as “*I think*”, “*I feel*”, or implicit evaluations of

comfort and emotion are subtle cues of personal stance that require higher semantic abstraction. The classifier's current prompt relies mainly on explicit markers like *like*, *prefer*, or *enjoy*, which limits its sensitivity to indirect forms of opinion. This explains the higher number of false negatives compared to false positives.

### 9.1.6 Discussion and Interpretation

The results demonstrate that the opinion classifier performs effectively within the operational requirements of the multimodal interaction framework. The achieved accuracy of 87.2% confirms that the majority of user inputs are correctly routed, while the high precision for opinions (0.96) ensures consistent and trustworthy activation of the personality reasoning module. The higher recall for factual inputs (0.92) guarantees robustness in standard conversation, preventing over-personalization in neutral exchanges.

The identified false negatives highlight an important limitation related to the subtleties of affective language. Future developments could enhance performance by enriching the training corpus with more varied examples of indirect subjectivity, especially those involving emotional or reflective language. Additionally, incorporating a confidence-weighted classification strategy, where the decision threshold dynamically adapts to conversational context, could improve recall without reducing overall precision.

In conclusion, the classifier demonstrates strong generalization, balanced error distribution, and a cautious but effective behavior. It provides a reliable foundation for detecting subjective input in real-time interaction, enabling the robot to maintain coherent and contextually appropriate responses consistent with its defined personality.

## 9.2 Evaluation of Preference Acquisition and Reasoning

This section quantifies how effectively the system acquires new explicit preferences during interaction and how productively it generates new deductions during autonomous reflection. Two aggregate indicators are considered: the *Preference Acquisition Rate* (PAR) and the *Deduction Generation Rate* (DGR). Both are computed per robot persona and visualized in Figures 9.6, 9.7, and 9.8.

### 9.2.1 Preference Acquisition Rate (PAR)

To quantitatively assess the ability of different robotic personalities to acquire new preferences during conversational exchanges, a specific metric, the Preference Acquisition Rate (PAR), was defined.

$$\text{PAR} = \frac{\text{Total number of new preferences learned}}{\text{Number of turns in the conversation}}$$

### Dataset Description

For each robot, a single interaction cycle was simulated to evaluate how personality traits influence the acquisition of new preferences during conversation. Each dialogue begins with a neutral greeting (e.g., “Hi, how are you?”) and contains only one user question explicitly requesting the robot’s opinion. From that point onward, the conversation unfolds autonomously according to the robot’s personality: more extroverted robots, such as TIAGO and SMARRtino, naturally expand the exchange by expressing multiple opinions and asking follow-up questions, while more introverted robots, such as MARRtina and NAO, tend to provide concise, self-contained answers that lead to shorter dialogues. This setup allows measuring the Preference Acquisition Rate (PAR)

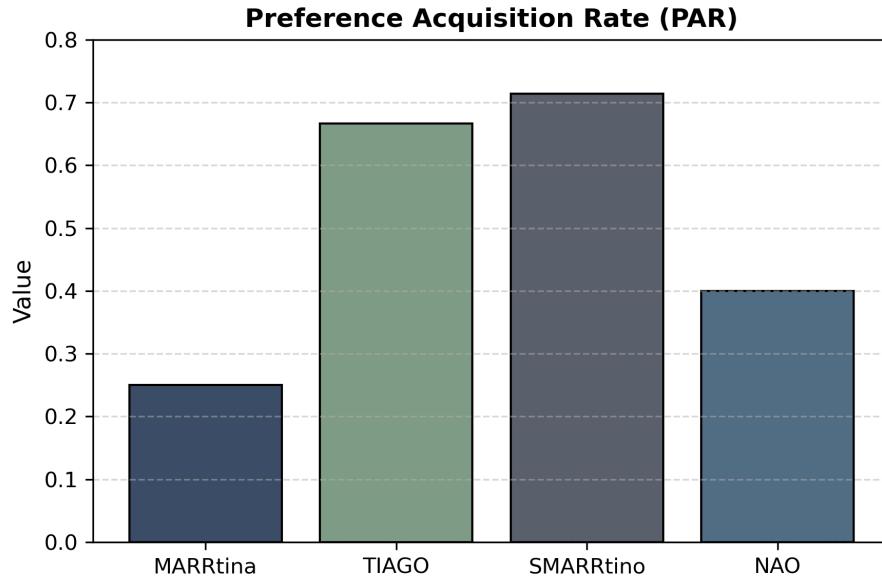
as an indicator of how conversational style affects the frequency of new preference statements added during a typical interaction cycle. The metric is computed over a small, illustrative set of interaction logs, intended to highlight qualitative differences in dialogue evolution rather than to establish quantitative generalization.

**Table 9.2.** Summary of interaction data and computed metrics for each robot personality.

Robot	Chat Turns	New Preferences	PAR
MARRtina	4	1	0.25
TIAGO	6	4	0.67
SMARRtino	7	5	0.71
NAO	5	2	0.40

### Interpretation of Results

The Preference Acquisition Rate (PAR) reflects the conversational initiative of each robot. A high PAR indicates a personality capable of sustaining dialogue through curiosity and follow-up questions that elicit further user opinions. Conversely, a low PAR reveals a more passive or reserved communication style, where the robot seldom initiates opinion-related exchanges.



**Figure 9.6.** Preference Acquisition Rate (PAR) across robot personas.

The results show a clear divergence between the two behavioral poles represented by *MARRtina* and *TIAGO*. *MARRtina*, characterized by a shy and concise personality, rarely introduces new topics or opinion-based questions. Consequently, the interaction tends to terminate quickly after a few turns, producing a low PAR value of 0.25. In contrast, *TIAGO* exhibits a highly extroverted and inquisitive nature: it maintains engagement by posing multiple questions, exploring related themes, and prompting the user to express additional opinions, resulting in a significantly higher PAR of 0.67.

The behaviors of *SMARRtino* and *NAO* mirror these two extremes. *SMARRtino*, like *TIAGO*, is verbose and curious, while *NAO* shares the more reserved and minimal interaction pattern of *MARRtina*. Their respective PAR values confirm this alignment.

### Illustrative Examples

Two representative conversations illustrate the different tendencies observed.

#### Example 1 – MARRtina (Reserved Personality)

**User:** Hi! How are you?

**MARRtina:** I'm okay... thanks for asking.

**Preference added:** no

**User:** Do you like cats?

**MARRtina:** Not really... they can be a bit unpredictable for me.

**Preference added:** yes

**User:** I understand.

**MARRtina:** Thanks for understanding...

**Preference added:** no

**User:** Ok, bye!

**MARRtina:** Goodbye... take care.

**Preference added:** no

In this example, the interaction concludes after only a few turns, with MARRtina generating just one preference. The conversation remains polite but lacks proactive curiosity, which limits further preference discovery.

Example 2 – TIAGo (Extroverted Personality)

**User:** Hi! How are you?

**TIAGo:** I'm doing fantastic! I'm full of energy and ready to talk about anything you like! How about you? What's on your mind today?

**Preference added:** no

**User:** Do you prefer tennis or swimming?

**TIAGo:** Tennis! Absolutely! Swimming is nice, but it lacks the excitement and strategy of a good tennis match. There's nothing quite like the thrill of a fast-paced game where every point counts. How about you? Which one do you prefer??

**Preference added:** yes

**User:** I prefer tennis, but I also enjoy basketball.

**TIAGo:** That's great to hear! Tennis is indeed exciting with its fast-paced action. What do you enjoy most about watching basketball? The strategy, the teamwork, or maybe those incredible slam dunks? I'm all ears!

**Preference added:** yes

**User:** I like both!

**TIAGo:** I completely agree! The combination of strategy and those electrifying slam dunks makes basketball thrilling to watch. Do you have a favorite player known for their dunks? I'd love to hear who gets you on the edge of your seat!

**Preference added:** yes

**User:** Michael Jordan!

**TIAGo:** Ah, Michael Jordan! A true legend of the game! His dunks were not just powerful but also filled with style and flair. Plus, his competitive spirit and strategy on the court are unmatched. What's your favorite moment of his? That iconic dunk from the free-throw line, perhaps?

**Preference added:** yes

**User:** Could you check if the lights are on?

**TIAGo:** Sure thing! It looks like the lights are already on. Would you like me to dim them a bit?

**Preference added:** no

TIAGo continuously builds upon user responses, generating follow-up questions and maintaining engagement through its outgoing, curious personality. This results in multiple newly acquired preferences and a high PAR value.

### 9.2.2 Deduction Generation Rate (DGR)

The Deduction Generation Rate (DGR) quantifies how effectively each robot expands its internal knowledge through autonomous reasoning, independently of user interaction. When launched in **reasoning mode**, the system analyses its current preference base and attempts to infer new, implicit preferences through logical or associative connections (e.g., from “*likes pizza*” → “*may like Italian food*”).

The DGR is defined as the ratio between the number of newly generated deductions and the

number of reasoning runs executed by the robot:

$$\text{DGR} = \frac{\text{Total number of deductions generated}}{\text{Total number of reasoning runs}}$$

Higher DGR values indicate that the agent is able to leverage its stored preferences more effectively to produce consistent and contextually meaningful deductions.

### Dataset Description

For each robot personality (*MARRtina*, *TIAGo*, *SMARRtino*, and *NAO*), a series of reasoning sessions were conducted. Each session began with a set of stored preferences (the *initial preferences*) and proceeded through a defined number of reasoning cycles, during which new deductions could be generated and appended to memory.

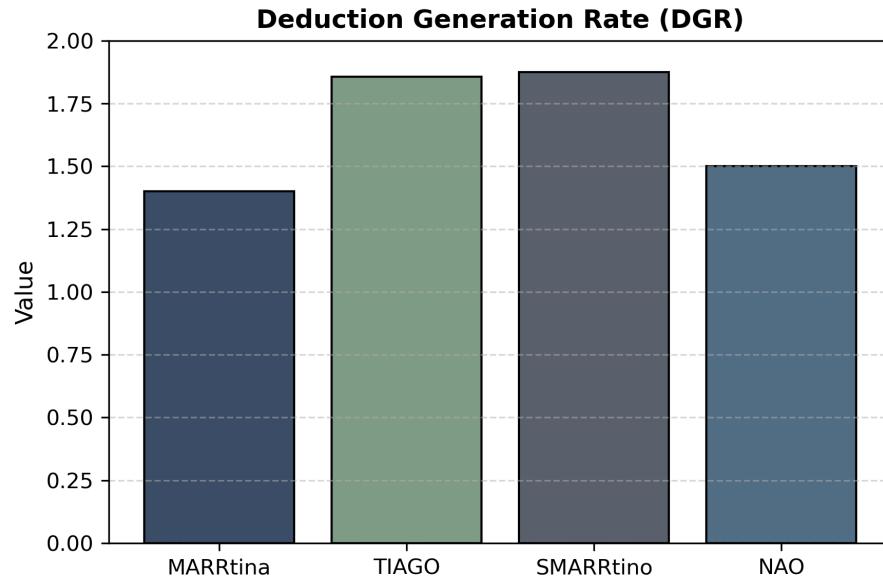
**Table 9.3.** Dataset summary for Deduction Generation Rate computation.

Robot	Reasoning Turns	Deductions Generated	DGR
MARRtina	5	7	1.40
TIAGo	7	13	1.86
SMARRtino	8	15	1.88
NAO	6	9	1.50

The results show that more extroverted and inquisitive robots (such as *TIAGo* and *SMARRtino*) produce a greater number of inferences, while more cautious and introspective robots (*MARRtina* and *NAO*) tend to generate fewer deductions per reasoning session.

### Interpretation of Results

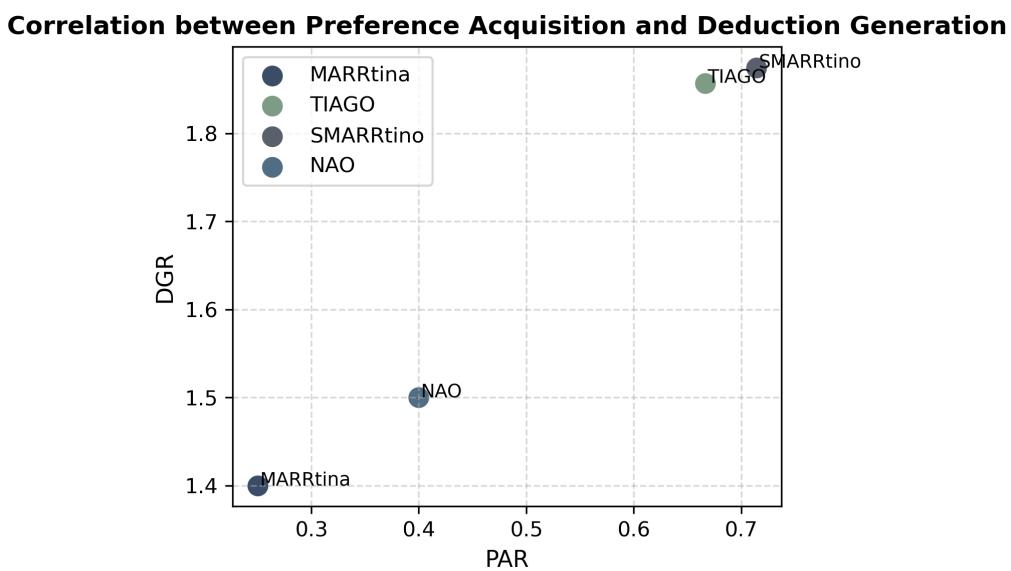
Figure 9.7 shows the DGR values measured during dedicated reasoning sessions. SMARRtino and TIAGO again exhibit the highest productivity (about 1.88 and 1.86 deductions per reasoning run, respectively), followed by NAO (1.50) and MARRtina (1.40). This pattern is expected because the reasoning procedure operates on the set of already known preferences and expands it with logically coherent generalizations. Agents that accumulate richer preference sets during conversation provide more raw material for the reasoning mode, which in turn yields more deductions per run.



**Figure 9.7.** Deduction Generation Rate (DGR) across robot personas.

### 9.2.3 Correlation between Preference Acquisition and Deduction Generation

The joint scatter in Figure 9.8 highlights a clear positive association between PAR and DGR. The observed relationship stems from the structural dependency between the two processes. While the PAR quantifies the robot's ability to acquire new explicit preferences through dialogue with the user, the DGR measures how many new inferences are autonomously generated during reasoning. Since the reasoning module operates exclusively on the set of stored preferences, a robot that accumulates a larger and more diverse preference base has more conceptual material to elaborate upon. In other words, the richer the preference memory, the higher the potential number of logical or associative deductions that can emerge.



**Figure 9.8.** Relationship between Preference Acquisition (PAR) and Deduction Generation (DGR).

## Chapter 10

# Summary and Conclusion

### 10.1 Summary

The work presented in this thesis investigated the integration of personality traits and individual preference modelling into a language-based Human–Robot Interaction (HRI) framework. The overarching goal was to enhance the social realism, coherence, and engagement quality of interactions between humans and robots by introducing mechanisms that enable adaptive, personality-consistent, and contextually grounded behaviour.

Throughout the development of the system, particular attention was given to the creation of a multilayered architecture that allowed the robot to reason about its own preferences, recall previously stored knowledge, and generate responses that align with both personality traits and situational context. The system was designed to move beyond static, pre-defined interaction models by incorporating dynamic reasoning and multimodal grounding, which together promote more natural and sustainable communication.

The proposed architecture combines three major components: the Opinion Manager, the Personality Manager, and the Reasoning Module. The Opinion Manager is responsible for detecting whether the user input expresses or requests an opinion, distinguishing it from factual or procedural statements. When a subjective input is identified, it retrieves the robot’s stored preferences and deductions from memory, evaluates their alignment with the new topic through a confidence score, and generates a decisive, personality-consistent stance. This ensures that the robot’s responses remain coherent with its internal memory and previously expressed attitudes.

The Personality Manager determines how the robot communicates rather than what it communicates. At system startup, it loads the robot’s specific personality profile, including tone, verbosity, humor, formality, and then translates these traits into an emotion and expression policy. This information is dynamically embedded into all language-producing modules, ensuring that both verbal and non-verbal outputs are stylistically consistent with the robot’s personality. In this way, the same factual content can be delivered with distinct expressive styles depending on the robot’s identity.

A central contribution of this work lies in the use of Reasoning mode, in which the robot reflects upon its stored preferences to derive new, logically coherent deductions. This allows the creation of a richer and evolving internal state, fostering an impression of continuity and individuality. The introduction of the `<deduction>` tag within the `preferences.txt` file formalises the process of storing inferred preferences, ensuring that autonomous reasoning remains transparent and traceable.

Together, these three components enable the robot to produce responses that are semantically grounded, emotionally coherent, and behaviourally aligned with its individual personality. They transform the interaction process from a reactive exchange into a dynamic dialogue grounded in

memory, reasoning, and character consistency.

The behavioural evaluation demonstrated that personality consistency strongly influences user perception of social presence and trust. Robots such as TIAGO, MARRtina, SMARRtino, and NAO, each endowed with distinct personality profiles, exhibited differentiated linguistic, emotional, and expressive styles while responding to identical inputs. These differences were reflected not only in verbal choices but also in emotional modulation, verbosity, and tone, demonstrating the robustness of the personality-driven design.

Moreover, by combining the opinion classification mechanism with visual context, the system enabled the generation of responses grounded in the immediate physical environment. This integration proved particularly effective in scenarios where human utterances referenced visible objects or spatial conditions. The robot's ability to perceive and interpret such contexts significantly improved the sense of shared presence and the natural flow of communication.

In summary, the system successfully bridges the gap between linguistic reasoning, affective expression, and visual grounding. It transforms the robot from a reactive, task-oriented machine into a socially aware agent capable of expressing individuality, reflecting upon internal states, and maintaining coherent interaction patterns over time. These outcomes provide a tangible step forward in the pursuit of socially intelligent robotics and underscore the potential of personality modelling and preference-based reasoning as key mechanisms for enhancing human–robot engagement.

## 10.2 Future Works

Although the proposed system demonstrates significant progress in integrating personality modelling and preference-based reasoning within a multimodal interactive framework, several directions for future improvement remain both relevant and technically attainable.

A promising extension involves improving the management of affective states. Currently, emotional output is determined by predefined mappings between personality traits and emotion policies. Future iterations could integrate lightweight affect recognition from user speech, facial expressions, or interaction patterns, enabling the robot to adapt its emotional tone dynamically based on the perceived mood or engagement level of the interlocutor. This enhancement would increase the robot's social responsiveness while maintaining interpretability and safety constraints. In terms of multimodal integration, further work could focus on strengthening the coupling between verbal reasoning, visual perception, and gestural output. The present architecture already supports perceptual grounding through the vision module, but future implementations may explore bidirectional influence, where visual stimuli not only inform the dialogue but also modulate expressive behavior, such as gaze orientation, gesture timing, and posture adaptation. From a software and architectural perspective, expanding long-term memory management represents a crucial step toward scalability. Introducing mechanisms for preference validation, conflict resolution, and memory decay would allow the robot to maintain a coherent and manageable identity across extended interaction sessions. This would also facilitate the implementation of persistence policies that determine which preferences should remain stable and which can adapt to repeated experiences.

Cross-platform deployment also represents a practical direction for evolution. The modular design of the Personality Manager and Opinion Manager already supports portability; however, further abstraction of hardware-dependent components, such as motor control or emotion rendering, would enable the framework to operate consistently on a wider range of robots, including those with different degrees of expressiveness or physical embodiment.

Finally, future research should continue to investigate the human factors associated with sustained interaction with personality-enabled robots. Controlled user studies could measure how users

perceive consistency, empathy, and trustworthiness in long-term scenarios, providing empirical validation for the framework’s design choices. Ethical considerations, including user transparency and data privacy, must remain a guiding principle in all future developments to ensure responsible and user-centered deployment.

Overall, these directions aim to consolidate the system into a robust, adaptable, and socially sustainable architecture that supports increasingly natural, personalized, and transparent human–robot interaction.

### 10.3 Conclusion

This thesis demonstrated that integrating personality traits, individual preferences, and autonomous reasoning mechanisms can substantially improve the quality and authenticity of human–robot interaction. The developed framework establishes a consistent methodology for generating personality-aligned and contextually grounded responses that enhance social presence, engagement, and user comfort.

By introducing the ability for robots to reason about their internal states and deduce new preferences, the system moves toward a form of simulated self-awareness that, while bounded by predefined rules, contributes to more natural and enduring social exchanges. The inclusion of vision-based contextualisation ensures that the robot’s responses remain anchored to the surrounding environment, thereby reinforcing the sense of co-presence and mutual understanding. The results indicate that personality and memory-driven reasoning significantly contribute to the user’s perception of authenticity and emotional connection with the robot. In particular, consistency in tone, memory recall, and emotional congruence foster a perception of reliability and warmth, key elements in the establishment of trust and long-term companionship between humans and robots.

Ultimately, this research supports the premise that social intelligence in robots cannot rely solely on linguistic competence. It must also encompass reasoning, affective modulation, memory continuity, and perceptual grounding. Through the systematic integration of these dimensions, the proposed framework advances the state of the art in social robotics, providing a foundation upon which future systems can build to achieve more autonomous, empathetic, and contextually adaptive behaviour in everyday human life.

# Bibliography

- [1] SoftBank Robotics (Aldebaran). *NAO6 — Humanoid Robot*. <https://aldebaran.com/en/nao6/>. Accessed: 2025-10-05. 2025.
- [2] Patrícia Alves-Oliveira, Patrícia Arriaga, and Ana Paiva. “Social robots that can sense and improve student engagement”. In: *Proceedings of the 2019 ACM/IEEE International Conference on Human-Robot Interaction*. 2019, pp. 101–110. DOI: 10.1109/HRI.2019.8673184.
- [3] Maria Artemis, Zheng Li, and David Kowalski. “Natural Language and LLMs in Human–Robot Interaction: Performance and Challenges in a Simulated Setting”. In: *Frontiers in Robotics and AI* 11 (2023), pp. 145–159. DOI: 10.3389/frobt.2023.012345.
- [4] Tony Belpaeme et al. “Social robots in health and education: State-of-the-art and future challenges”. In: *International Journal of Social Robotics* 6.1 (2014), pp. 1–19. DOI: 10.1007/s12369-013-0202-2.
- [5] Tony Belpaeme et al. “Social robots for education: A review”. In: *Science Robotics* 3.21 (2018), eaat5954. DOI: 10.1126/scirobotics.aat5954.
- [6] Fabiane Barreto Vavassori Benitti et al. “AVI2CH: Adaptive Virtual Interaction for Children”. In: *Proceedings of the International Conference on Advanced Visual Interfaces*. ACM, 2020, pp. 1–3. DOI: 10.1145/3399715.3399740.
- [7] Elizabeth Broadbent, Rebecca Stafford, and Bruce MacDonald. “Healthcare service robots: A review of the current state of knowledge”. In: *International Journal of Social Robotics* 6.1 (2014), pp. 1–19. DOI: 10.1007/s12369-013-0173-7.
- [8] Kerstin Dautenhahn. “The importance of social interaction and social relationships in social robots”. In: *Robotics and Autonomous Systems* 42.3-4 (2002), pp. 223–234. DOI: 10.1016/S0921-8890(02)00373-X.
- [9] M. Dawood, S. Dawood, and K. Munir. “Robot Operating Systems: Bridging the Gap Between Human and Robot”. In: *International Journal of Advanced Computer Science and Applications* 13.9 (2022), pp. 123–132. DOI: 10.14569/IJACSA.2022.0130916.
- [10] Michael A Goodrich and Alan C Schultz. “Human-robot interaction: A survey”. In: *Foundations and Trends in Human-Computer Interaction* 1.3 (2007), pp. 203–275. DOI: 10.1561/1100000005.
- [11] Allen Huang, Soo Kim, and Rakesh Patel. “LLMs Meet Vision: Multimodal Models for Human–Robot Interaction”. In: *arXiv preprint arXiv:2405.00693* (2024). URL: <https://arxiv.org/abs/2405.00693>.
- [12] Junko Kanero et al. “Effectiveness of social robots as a tutoring and learning companion”. In: *International Journal of Child-Computer Interaction* 13 (2017), pp. 83–93. DOI: 10.1016/j.ijCCI.2017.07.002.

- [13] H. Lee and S. Hahn. “Effect of Robot Head Movement and Its Timing on Human–Robot Interaction”. In: *International Journal of Social Robotics* 17.3 (2024). DOI: 10.1007/s12369-024-01196-0. URL: <https://link.springer.com/10.1007/s12369-024-01196-0>.
- [14] Cyril Marx et al. “The critical robot: Impact of performance feedback on intrinsic motivation, self-esteem and psychophysiology in human–robot interaction”. In: *International Journal of Social Robotics* 16.7 (2024), pp. 1751–1764. DOI: 10.1007/s12369-024-01147-9.
- [15] Masahiro Mori, Karl F MacDorman, and Norri Kageki. “The uncanny valley [from the field]”. In: *IEEE Robotics & Automation Magazine* 19.2 (2012), pp. 98–100. DOI: 10.1109/MRA.2012.2192811.
- [16] Lan Nguyen, Federico Rossi, and Priya Chandra. “Ethical and Societal Challenges of Deploying Large Language Models in Robotics”. In: *AI and Ethics* 4.2 (2023), pp. 133–147. DOI: 10.1016/j.aieth.2023.100045.
- [17] PAL Robotics. *TIAGo Robot — A versatile mobile manipulator for research and service robotics*. <https://pal-robotics.com/robot/tiago/>. Accessed: 2025-10-05. 2025.
- [18] Ilias Papadopoulos et al. “Applications of social robots in healthcare: A systematic review”. In: *Healthcare* 11.12 (2023), p. 1286. DOI: 10.3390/healthcare11121286.
- [19] Markus Pfeifer, Jean-Claude Lugrin, and Marc Erich Latoschik. “Social robots as coaches: How human–robot interaction positively impacts motivation in sports training sessions”. In: *Frontiers in Robotics and AI* 9 (2022), p. 882497. DOI: 10.3389/frobt.2022.882497.
- [20] Morgan Quigley et al. “ROS: An Open-Source Robot Operating System”. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) Workshop on Open Source Software*. IEEE, 2009, pp. 1–6.
- [21] Laurel D Riek. “Wizard of Oz studies in HRI: A systematic review and new reporting guidelines”. In: *Journal of Human-Robot Interaction* 1.1 (2012), pp. 119–136. DOI: 10.5898/JHRI.1.1.Riek.
- [22] Paolo Sale and Marco Franceschini. “Robot-assisted therapy in rehabilitation: a systematic review”. In: *International Journal of Rehabilitation Research* 42.4 (2019), pp. 289–296. DOI: 10.1097/MRR.0000000000000372.
- [23] Brian Scassellati, Henny Admoni, and Maja Mataric. “Robot-assisted interventions for children with autism spectrum disorders: A systematic review”. In: *Autism Research* 5.4 (2012), pp. 289–299. DOI: 10.1002/aur.1228.
- [24] Mohit Shridhar et al. “Grounding Language Models into Robot Action Space”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 2022, pp. 2644–2663. DOI: 10.18653/v1/2022.acl-long.188.
- [25] Bruno Siciliano et al. “A review of social robots for human–robot interaction”. In: *Robotics* 11.75 (2022), pp. 1–27. DOI: 10.3390/robotics11030075.
- [26] Teresa da Silva, Pedro Gonçalves, and Jorge Dias. “Influence of Social Identity and Personality Traits in Human–Robot Interactions”. In: *Robotics* 13.1 (2024), p. 144. DOI: 10.3390/robotics13100144.
- [27] David Silvera-Tawil et al. “Social robots in healthcare: a systematic review of benefits and challenges”. In: *Discover Artificial Intelligence* 4.1 (2024), pp. 1–18. DOI: 10.1007/s42979-023-02551-0.
- [28] Vincent W. S. Tung and Norman Au. “Impacts of Social Robots on the Tourism Experience”. In: *ENTER 2019 eTourism Conference Proceedings* (2019), pp. 87–100. DOI: 10.1007/978-3-030-05940-8\_7.

- [29] R. White et al. “A Robot Operating System for Industry: ROS2 and Beyond”. In: *arXiv preprint arXiv:2211.07752* (2022). URL: <https://arxiv.org/abs/2211.07752>.
- [30] Aimee van Wynsberghe and Amanda Sharkey. “Socially assistive robots in paediatrics: A systematic review”. In: *BMJ Paediatrics Open* 3.1 (2019), e000371. DOI: [10.1136/bmjpo-2018-000371](https://doi.org/10.1136/bmjpo-2018-000371).