Received 10 March 2025, accepted 23 April 2025, date of publication 30 April 2025, date of current version 8 May 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3565918

RESEARCH ARTICLE

Large Language Models in Human–Robot Collaboration With Cognitive Validation Against Context-Induced Hallucinations

NADUN RANASINGHE[1], (Graduate Student Member, IEEE), WAEL M. MOHAMMED[1], KOSTAS STEFANIDIS[2], AND JOSE L. MARTINEZ LASTRA[1], (Member, IEEE)

[1] FAST-Laboratory, Faculty of Engineering and Natural Sciences, Tampere University, 33720 Tampere, Finland

[2] Faculty of Information Technology and Communication Sciences, Tampere University, 33720 Tampere, Finland

Corresponding author: Nadun Ranasinghe (nadun.ranasinghe@tuni.fi)

This work was supported in part by the Finnish Doctoral Program Network in Artificial Intelligence (AI-DOC), under the Grant VN/3137/2024-OKM-6; and in part by the European Union's Horizon Europe research and innovation program under grant agreement n° 101058589 which corresponds to the AI-Powered human-centred Robot Interaction for Smart Manufacturing (AI-PRISM) project.

ABSTRACT The recent leap in Large Language Models (LLMs) has paved the way for several research ideas. LLMs are employed not only for personal use but also in professional contexts to enhance human productivity at work. A significant area of research is human-robot collaboration (HRC), which focuses on developing methodologies for effective interaction between humans and AI-enabled machines. In this regard, exploitation of LLMs appears to be a practical approach. However, these models are susceptible to several limitations, including context-induced errors, the propagation of misleading information, and hallucinations. Such deficiencies impede the seamless application of LLMs in scenarios where a high degree of accuracy is essential. To address this issue, this study introduces a dual-agent system designed to validate the responses generated by LLMs. This novel system is integrated into a framework called "CogniVera", which facilitates collaborative tasks involving a collaborative robot (cobot) through vocal interactions. This initiative represents a significant advancement in HRC, enabling robots to communicate vocally with human operators during assembly tasks. To evaluate the feasibility of this approach, a focused case study will be conducted, concentrating on the human-robot collaborative task of box assembly utilizing vocal communication. The outcomes of this study are anticipated to yield valuable insights into the efficacy of the proposed dual-agent system in enhancing the reliability and performance of LLMs in practical applications.

INDEX TERMS Artificial intelligence (AI), human–robot interaction (HRI), large language models (LLM), conversational artificial intelligence, human–robot collaboration (HRC).

I. INTRODUCTION

In recent decades, there has been a significant advancement in research on Industrial Automation, leading to a transformative paradigm shift towards Industry 5.0. As articulated by the European Commission, this paradigm is founded on three fundamental pillars: Human-Centricity, Resilience, and Sustainability in the design and implementation of industrial technologies [1]. One core concept in the Human-Centricity pillar in Industry 5.0 is the Human-Robot Collaboration (HRC), which is a specific case within Human-Robot Interactions (HRI) [2]. As presented in [3], this involves collaborative robots (cobots) working synergistically with human operators, thereby integrating human cognitive capacities and creative skills with the accuracy and efficiency of cobots to enhance the execution of various tasks. However, until recently, interactions between cobots and humans during HRC tasks were limited to pre-defined gestures, basic commands, or programming interfaces, rendering real-time collaboration inefficient [4].

The associate editor coordinating the review of this manuscript and approving it for publication was Zijian Zhang.

Nevertheless, with the emergence of the new paradigm, research focus has shifted towards facilitating more intuitive and human-like communication, thereby fostering a more natural interaction between humans and cobots. The significance of this focus has been increasingly emphasized with the advent of Transformer Networks [5] in the AI domain. These networks have demonstrated the ability to train and develop advanced language models, commonly referred to as Large Language Models (LLM). These models exhibit a sophisticated understanding of natural language context, possess reasoning capabilities, and can adapt their performance based on incoming data, among other functionalities. Integrating these models into the HRC concept facilitates seamless and reliable natural language interactions between human operators and cobots, bringing this vision closer to reality.

The current body of literature thoroughly examines the integration of LLMs within HRC aimed at enhancing non-physical HRI through natural language communication [6], [7], [8], [9], [10]. However, numerous proposed approaches have encountered challenges that impede the establishment of natural and reliable continuous communication between humans and robots during the execution of complex tasks. This capability is vital within industrial contexts. [11]. A significant issue is the mitigation of hallucinations—erroneous outputs generated by LLMs [12] that can significantly compromise human-robot collaborative systems. In industrial settings, these errors can also arise from semantic ambiguities present in the environment; for example, a natural language-driven HRC may misinterpret verbal commands due to ambient industrial factors or the constraints of employed speech-to-text modules.

With the noticeable need for robust and redundant interfacing between humans and robots, this study introduces an innovative framework that integrates LLMs to address these challenges and enhance human-robot collaboration within industrial environments. As shown in Figure 1, the proposed framework involves the partnership of two LLM-based agents that function synergistically to identify and mitigate hallucinations, thereby ensuring the reliability and accuracy of the system outputs. This architecture features a continuous execution of complex tasks that require a series of sequential operations. Furthermore, it promotes effective and seamless interaction between cobots and human operators through advanced vocal communication, thus fostering a more efficient and intuitive collaborative experience in industrial settings. The proposed system is validated through a specific Human-Robot collaborative case study, which illustrates its effectiveness compared to the traditional integration of a single prompted agent. This agent utilizes a combination of features discussed in the existing literature in Section II, including feedback for decision-making, as introduced in Incoro [7].

Overall, the main contributions of our work are the following:

1) This study introduces a novel methodology designed to mitigate errors and hallucinations through an internal dialogue mechanism involving two LLM-based agents, referred to as Cognitive Response Verification (CRV).
2) The proposed methodology will be integrated into the new CogniVera framework, which will incorporate this methodology into a Human-Robot Collaboration system to facilitate collaborative tasks between humans and cobots through vocal communication.
3) The entire framework is validated through a well-established collaborative case study on box assembly, with the aim of evaluating its efficacy and reliability.

The remainder of this paper is organised as follows. Section II provides a comprehensive literature review that aims to illuminate the existing body of research and identify the specific research gap this study seeks to address. Section III provides a detailed description of the proposed architecture and outlines its key components and functionalities. In Section IV, we define the case study employed for validation purposes, including a thorough explanation of the validation testing methodology used in this research. Section V concludes the paper by synthesizing the findings derived from the validation process, discussing the implications of the proposed architecture, and providing recommendations for future research directions.

II. RELATED WORK

The integration of LLMs into HRC has emerged as a crucial focus in contemporary research. This focus is driven by the aim to improve natural language interactions between humans and cobots. However, this integration creates challenges, as LLM errors, commonly referred to as Hallucinations, along with other potential discrepancies, can significantly impact HRC systems. Accordingly, research efforts are simultaneously directed toward addressing these issues to ensure the development of reliable, error-free HRC systems. This section aims to review the existing literature thoroughly, define the innovations presented therein, and identify the gaps that this study seeks to address.

A. HUMAN-ROBOT COLLABORATIONS

Human-robot collaboration, a subcategory of Human-Robot Interaction, can be defined as a scenario in which one or more humans team up with one or more robots to work together towards a common purpose, performance goal, and approach for which they hold themselves mutually accountable [3]. This means that, in contrast to closed and enclosed robots that function independently according to human directives and programming, cobots operate in synergy with human operators to attain shared objectives while maintaining close physical proximity. In light of the increasing demand for greater flexibility and efficiency in manufacturing processes, driven by the paradigms of Industry 4.0 [13] and Industry 5.0 [1], the integration of HRC systems within industrial environments has begun to cause significant

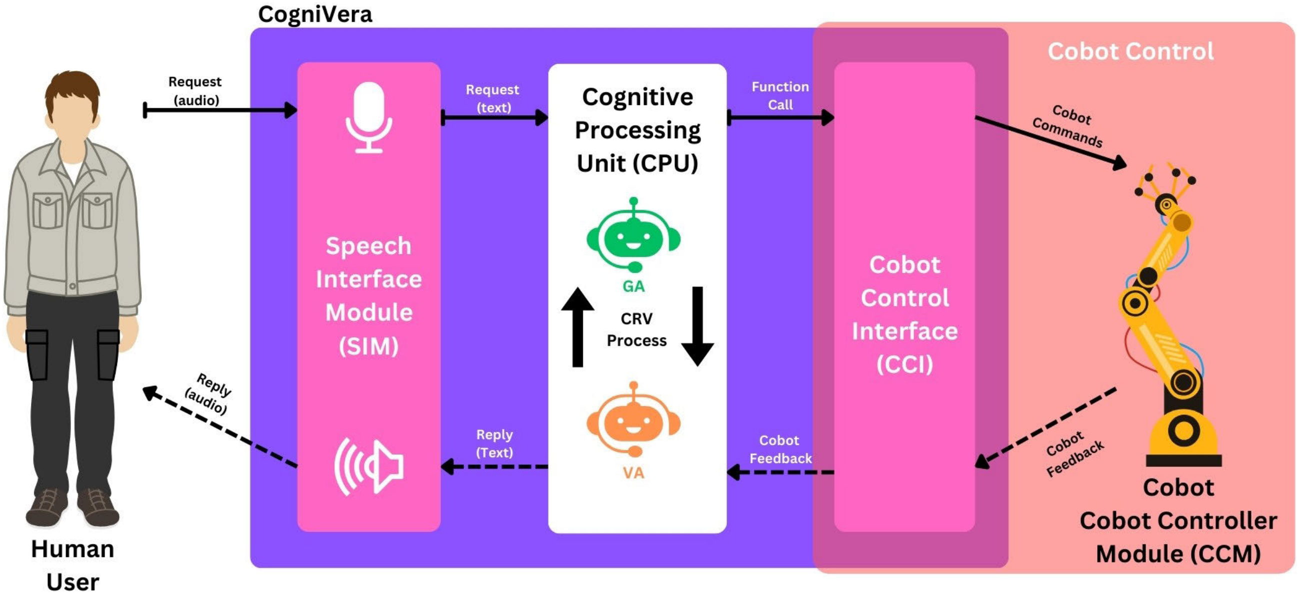


FIGURE 1. The CogniVera framework.

changes in the industry [14]. A thorough examination of the current literature on HRC within industrial contexts reveals that the applications of HRC can be classified into three distinct categories: collaborative transportation, collaborative assembly, and collaborative manipulation of objects [15]. Collaborative transportation refers to the movement of substantial and/or heavy objects that require additional assistance, wherein cobots may be employed [16], [17], [18]. Collaborative assembly refers to a scenario in which a human operator collaborates with a cobot to assemble or disassemble a specific object. This process requires the active participation of both parties [19], [20], [21], [22]. To effectively execute an assembly task collaboratively between a cobot and a human operator, it is essential to follow a sequential process. Each participant must assume responsibility for distinct steps within the sequence, thereby ensuring that the assembly process is completed accurately and that the final product is assembled to the required specifications. Establishing seamless communication between the cobot and human operator is crucial for facilitating the effective execution of a sequential process in a collaborative environment. Consequently, it is essential to employ verbal or non-verbal communication methods to enhance their interaction [23]. In the transition to Industry 5.0, which is characterized by a human-centric approach as one of its foundational pillars, there has been an increasing emphasis on the necessity for HRC systems to prioritize human-centric design. This evolution aimed to enhance the interaction between human operators and cobots, fostering a perception of interaction akin to that between humans. Such advancements are crucial for establishing a more effective relationship between humans and cobots, ultimately prioritizing operator health and safety while simultaneously addressing the industry's demands for efficiency and flexibility [24].

B. LARGE LANGUAGE MODELS IN HRC

LLMs are sophisticated computational models developed using an innovative Transformer architecture, fundamentally based on the self-attention mechanism [5]. These models possess the capability to comprehend natural language inputs and generate contextually relevant responses by sequentially predicting each token of the reply [25]. With the introduction of prominent LLMs such as GPT [26] and BERT [27], rapid development in this field has led to the emergence of a wide range of models, including notable examples such as LLaMA [28], Mistral [29], Claude [30], DeepSeek [31] and GPT-4 [32]. These models exhibit two primary features: in-context learning and Reinforcement Learning from Human Feedback (RLHF). In-context learning, as presented in [33], refers to the models' ability to comprehend the contextual significance of a given request, enabling them to generate responses with enhanced accuracy. This capability fosters a more human-like quality in their conversational and interactive functionalities. Additionally, the capacity to refine their performance over time by incorporating human feedback is encapsulated in the term RLHF [34]. This process signifies the models' ability to adapt and enhance their outputs based on external guidance. Consequently, these capabilities have ignited a renewed research focus on integrating large language models (LLMs) into existing industrial Human-Robot Collaboration (HRC) systems across a variety of applications. This integration enhances human-centered interaction between robots and human operators, promoting a more collaborative and intuitive working environment.

LLMs have been integrated using various methodologies in numerous applications. These methodologies involve directly utilizing a specifically fine-tuned model or modifying an existing proprietary or open-source model to meet specific requirements, employing prompt engineering and other related techniques. Liang et al. [6], proposed a framework that leverages LLMs to generate executable robot policy code from natural language commands. Their findings demonstrated that LLMs can be effectively employed for code generation and task planning in robotic systems. Task planning entails a systematic approach in which human operators articulate a complex task comprising multiple sub-tasks as a natural language request directed at LLMs. In response, LLM generates the corresponding sub-tasks coherently and sequentially, thereby facilitating execution in alignment with the original request. Liu et al. [35], introduced a framework leveraging LLMs designed for the task planning of intricate, long-horizon assignments. Furthermore, various methodologies have emerged that utilize LLMs for the planning of tasks pertinent to HRC [36], [37], [38].

LLM integration on Robot control represents a significant advancement in task planning. In this approach, LLMs are directly integrated with cobots to execute tasks based on language requests. The RT-2 model [39], which encompasses vision-language-action capabilities, excels in generating robotic control parameters and guiding robotic actions in response to such requests. Additionally, it demonstrates the ability to interpret novel commands and engage in basic reasoning processes, including categorizing objects and analyzing high-level descriptions. Zhu et al. [7], presented InCoRo, a system that leverages off-the-shelf LLMs as traditional robot controllers, enabling robots to respond to natural language instructions and adapt to various scenarios through a feedback loop mechanism supported by computer vision-based perception. Numerous methodologies have been proposed to enhance direct communication between humans and cobots [40], improve interactions for non-expert users [41], and increase the overall flexibility within this domain [42].

In summary, the existing literature demonstrates that LLMs are effectively integrated into HRC and are capable of performing complex, long-term tasks in response to natural language commands. However, a notable research gap exists in enhancing the human-centric nature of these systems, specifically through the development of more humanized, intuitive conversational interactions. Furthermore, it is essential to address safety concerns by mitigating hallucinations and resolving errors prevalent in industrial environments.

C. HALLUCINATIONS AND ERROR MITIGATION

In industrial environments, the successful and safe execution of tasks utilizing an HRC system requires assurance that the system is equipped to mitigate a comprehensive range of potential errors [43]. Within HRC frameworks that are advanced through the integration of LLMs, it is crucial to identify and address two primary categories of errors: the hallucinations produced by LLMs and the errors stemming from the industrial context, which may instigate system failures. Hallucinations can be defined as outputs generated by LLMs that, while appearing plausible and coherent, are ultimately unfaithful to the underlying data and present information that may be inaccurate or nonsensical [12]. These random outputs can disrupt the functioning of the HRC process and cobot, adversely affecting the system's safety and reliability. Errors prevalent in industrial settings that can lead to system failure can encompass various subtypes. However, in the context of this study, which focuses on large language model integrated Human-Robot Collaboration (LLM-HRC) systems, these errors primarily arise from noisy environments and inaccuracies associated with natural language requests. Furthermore, challenges in the comprehension of such requests by speech-to-text mechanisms significantly contribute to these errors.

The existing literature highlights a variety of techniques employed for error mitigation. Notable among these are the utilization of pre-trained LLMs and Visual Language Models(VLMs) for Robot Task planning and execution [39], [44], the application of prompting techniques such as chain-of-thought approaches for code generation in robot navigation [2], and the integration of perception data into LLM requests to enhance comprehension [35]. Furthermore, methodologies that incorporate control systems, including feedback loop systems [7], Bayesian networks [37], and multi-agent frameworks [40], are frequently discussed in the literature. However, upon reviewing the methodologies mentioned earlier, it becomes evident that these approaches either require substantial time investment, lack flexibility, exhibit complexity or are limited to error resolution rather than error mitigation. Substantial research opportunities have arisen regarding the robustness and error mitigation features that an HRC requires while implementing LLM as an interfacing mode.

III. APPROACH

Looking at the literature landscape, it is visible that the HRC domain lacks systems capable of executing collaborative tasks that involve sequential interactions between human operators and cobots. These systems must effectively mitigate the errors encountered during these processes, addressing both system-generated errors and misinterpretations, commonly referred to as hallucinations. CogniVera, short for Cognitive Verification, is an innovative framework that features a new process called "Cognitive Response Validation" (CRV). This framework allows an HRC system to ensure error-free interaction between the human operator and the cobot, facilitating effective collaboration. The CogniVera framework consists of three pivotal components: the Speech Interface System (SIM), Cognitive Processing Unit (CPU), and Cobot Control (CC). This section comprehensively examines these core components, detailing their interrelationships and how their synergistic interaction contributes to the framework's overall functionality.

A. SPEECH INTERFACE MODULE (SIM)

The Speech Interface Module is the primary module that directly interacts with a human operator through vocal conversations. Thus, its role is to ensure that the given Human audio-based requests are converted to the required text-based format for better understanding by the system and vice versa. The presence of this component allows flexibility in using any off-the-shelf or trained model for speech-text conversions in both directions, so that it can be selected according to the environment in which the HRC system exists. To ensure the optimal functionality of this component, the following requirements must be fulfilled during the implementation process.

- The SIM must establish connections with audio recording devices to collect audio data upon request. Subsequently, it converts the acquired audio data into a textual format that can be processed and comprehended by LLM agents.
- SIM must be integrated with audio output devices to facilitate the dissemination of vocal responses generated by LLM agents. This integration entails the conversion of textual replies into audible responses that human operators can comprehend.
- The SIM system should facilitate seamless dialogue that dynamically interweaves operator requests with system responses throughout the collaborative process.

This component is designated as a "module" due to the Cognivera framework's inherent flexibility. This framework facilitates the adaptation of interaction modes between humans and cobots based on the specific application context. It is important that the requirements mentioned earlier are consistently met, irrespective of the chosen method of interaction.

B. COGNITIVE PROCESSING UNIT (CPU)

The Cognitive Processing Unit (CPU) is the central component of the proposed framework and functions similarly to the human brain. Within this framework, two LLM-based agents engage in an internal dialogue to implement the Cognitive Response Validation (CRV) process. This mechanism is designed to ensure that all the responses generated in response to operator requests are accurate, executable, and error-free. This section provides a concise overview of the two principal agents within this component. Subsequently, the innovative CRV methodology and structured prompting employed to optimize their functionality are discussed in detail.

1) GENERATION AGENT (GA)

The generation agent is a principal entity based on a large language model (LLM) that processes input from human operators and produces pertinent responses, including function calls for robotic execution. Similar to the LLM agents introduced in the literature, such as InCoro [7], this particular agent is also designed to receive feedback from the robot and directives from the validation agent during internal communications. Consequently, the generation agent will be structured to fulfil the following responsibilities:

1) Receive and process natural language requests from human operators, generate appropriate responses, and issue necessary function calls for the robot as required.
2) Engage with and critically analyze internal communications from the Validation agent, addressing and rectifying any errors identified in the generated output as necessary.
3) Analyze and process the feedback provided by the cobot through the CC during task execution to ensure that its actions align with the specified request, or determine whether the subsequent sequential action should be executed.
4) To effectively receive and process information pertaining to the prior state of the system during the generation of responses.
5) While LLMs exhibit considerable flexibility in addressing a wide range of requests, it is crucial to ensure that cobots, which are designed for specific tasks, invoke only pre-defined functions. The LLM-based GA must refrain from accessing any functions outside these established parameters to prevent potential disruptions to the overall system.

2) VALIDATION AGENT

The Validation Agent (VA) was a novel component introduced in this study. It aims to enhance the internal communication with the Generation Agent to rectify errors prior to the transmission of outputs from the CPU through the CRV process. This agent accepts both the input provided and the output generated by the Generation Agent to evaluate whether the generated output appropriately corresponds to the input request. Following this assessment, VA produces an accuracy score and delivers textual feedback to the GA when a modification of the response is warranted. Furthermore, the VA produces state information derived from the specified input request and generated output. This information enables the GA to comprehend the current status of tasks more effectively. The responsibilities of the Validation Agent can be defined as follows: (a)

1) Receive and analyze the input request along with its corresponding output response generated by the GA, ensuring that they align correctly and contain no errors.
2) Assign an accuracy score to each request-response pair received in order to validate their correctness and reliability.
3) In the event that errors are identified within the request-response pair, it is essential to generate coherent textual feedback to facilitate internal communication with GA for the purpose of rectifying the issue.
4) Upon confirming the accuracy of the request-response pair, it is essential for the VA to generate the current state of the process in order to inform the GA appropriately.

3) COGNITIVE RESPONSE VALIDATION (CRV)

Guided by the methodology presented in [45], which utilizes a score-based self-reflection approach to alleviate hallucinations in question-answering tasks, particularly within the medical domain, Cognitive Response Validation (CRV) has been developed to improve accuracy through the systematic identification and filtering of errors. This is accomplished by establishing an internal dialogue between two agents: the GA, responsible for addressing all requests, and the VA, which evaluates the accuracy of the generated responses and ascertains that the requests can be executed without risk of unintentional harm to the HRC system. This framework effectively mitigates the hallucinations that are often associated with LLMs during HRC tasks. The internal dialogue of the CRV can be defined as an iterative process. This guarantees the safety and reliability of the system execution between the user and the cobot. The steps are as follows:

1) GA generates the required output response for the given input response.
2) VA assesses the GA output using the provided input and calculates an accuracy score (defined in section III-B.4).
3) If $K \leq K_{AVG}$, the VA provides feedback to the GA concerning the identified errors, after which the GA revises and regenerates the required output. In contrast, if $K > K_{AVG}$, the generated output is transmitted to the CC and SIM for the continuation of the process.
4) The steps 1, 2 and 3 will be repeated n times until $K > K_{AVG}$.

As shown in algorithm 1, the constant value n represents a specific finite number. It is utilised to prevent infinite loops that may arise owing to the hallucinations experienced by LLM-based agents during the CRV process. This precautionary measure is crucial to ensure the safety of the framework. Algorithm 1 and the sequence diagram shown in Figure 2 also provide a better understanding of this novel methodology.

Algorithm 1 Cognitive Response Validation Process

```
Initialize parameters:
$K\_AVG \leftarrow x$ {Set accuracy threshold}
$MAX \leftarrow n$ {Set maximum allowed iterations}

Function CRV_Process($input$):
    $iter \leftarrow 0$ {Initialize iteration counter}
    while $iter < MAX$ do{Step (4)}
        {Step (1)}
        $output \leftarrow GA.generate\_output(input)$
        {Step (2)}
        $(K, feedback) \leftarrow VA.evaluate(output, input)$
        if $K > K\_AVG$ then
            return $output$ {Accept output}{Step (3)}
            break $output$ {Terminate Loop}
        else
            $input \leftarrow feedback$ {Step (3)}
            $iter \leftarrow iter + 1$ {Increment iteration counter}

            continue {Restart loop}
    end while
    raise Error("CRV Process failed: Maximum iterations reached without acceptable output")
```

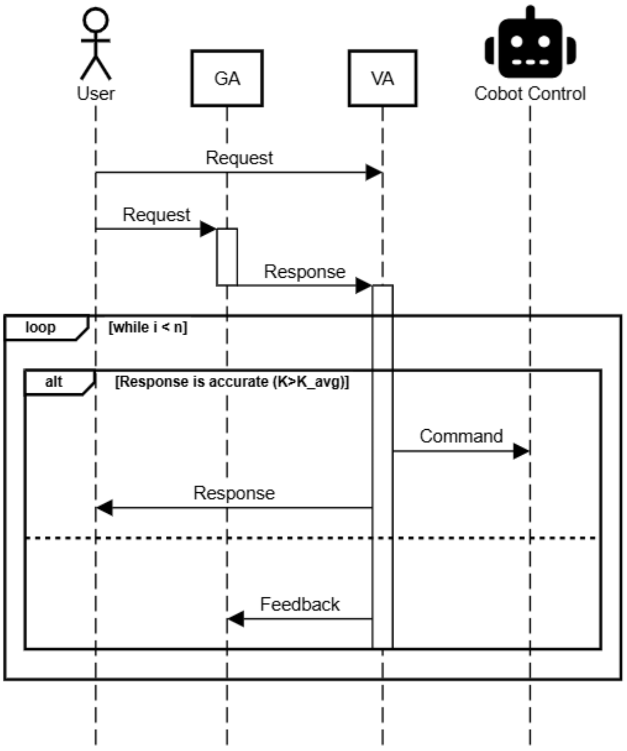


FIGURE 2. The CRV process sequence diagram.

4) ACCURACY SCORE

The VA assigns each request-response pair an accuracy score to differentiate between erroneous and accurate responses. This score is utilised to assess whether the pairs can be sent as is or if they require modification to rectify any identified errors. The derivation of this score, along with its associated rules, can be expressed as follows:

The accuracy score K should be defined between a specified range $[K_{MAX}, K_{MIN}]$, where the threshold value K_{AVG} is defined as shown in the Equation 1.

$$K_{AVG} = (K_{MAX} + K_{MIN})/2 \tag{1}$$

The accuracy score K is determined following a specific set of rules:

1) If $K = K_{MAX}$, the generated output **satisfies** the specified input criteria **accurately**, thereby ensuring the integrity of the system and successful completion of the task.
2) If $K_{MAX} > K > K_{AVG}$, the generated output is identified as optimizable and the given input is

deemed to be somewhat inaccurate; however, further optimization is deemed **unnecessary**, as it will exert minimal impact on both the system and the task at hand.

3) If $K_{MIN} < K \leq K_{AVG}$, the output has been identified as optimizable, while the provided input appears to exhibit inaccuracies; therefore, further optimization is **necessary** to mitigate any potential impact on the system.
4) If $K = K_{MIN}$, the generated output and input are considered to be **inaccurate**, resulting in a substantial impact on the overall system.

Textual feedback is generated only when the system output needs to be modified, specifically when $K \leq K_{AVG}$. This effectively minimizes the utilization of unnecessary tokens by LLMs.

5) STRUCTURED PROMPTING

The system must possess the capability to satisfy several fundamental requirements. The system should enable direct internal communication between the two agents to facilitate error mitigation and permit direct interactions with the robot, including issuing commands and receiving feedback. Furthermore, it is essential to execute tasks sequentially to address a specified complex request autonomously. First, the inputs and outputs of the CPU should be easily manipulated by other system components. Second, agents must clearly understand the specific actions they should undertake in response to each received input. To this end, a structured prompting technique was implemented to guide agents effectively. The defined prompting structure is designed to adhere to a JavaScript Object Notation(JSON) [46]based format, wherein each object represents a specific parameter of the agent. These parameters encompass the agent's description, functions available for invocation, the input format the agent receives, and output format that the agent is expected to generate, as illustrated in Listing 1.

```
Act as the following agent:
{
    "Name": "Name of the Agent",
    "Description": "Description of the
                    agent including
                    all the defined
                    responsibilities",
    "Functions": "Function list that
                  can be called",
    "Input": "Input format received by
              the agent",
    "Output": "Output format generated
               by the agent."
}
```

LISTING 1. Prompt format for an agent.

Utilizing a structured methodology for the inputs provided and outputs generated by agents, as illustrated in Listing 2, facilitates easier manipulation and enhances the clarity of information conveyed to other system components.

```
{
    "OP": {"Reply": "The
                     conversational
                     reply to the
                     Message, should
                     be in spoken
                     language ",
           "Function":
           {"Name": "Name of the
                     function to be
                     called to
                     satisfy the
                     user's request.",
            "Params":
            {"Name of the parameter":
            "Value of the parameter",
            }}}
}
{
    "IP": {"Type": "The type of
                    message",
           "Data": "The contents
                    of the message"},
           "State": "State of the
                     system and process"
}
```

LISTING 2. Sample Input and Output format.

C. COBOT CONTROL

Cobot Control functions as an intermediary between the overarching framework and the cobot. Its principal objective is to facilitate the efficient transfer of data between the framework and the cobot, enabling the execution of actions within the cobot. In addition, it plays a crucial role in transmitting feedback from the cobot back to the CPU. This component comprises two integral modules: the Cobot Control Interface (CCI) and the Cobot Controller Module (CCM). CCI is embedded within the framework to facilitate data transfer between the cobot and the framework, while the CCM is integrated into the cobot controller. CCM is responsible for receiving data from the framework, executing the necessary actions for the cobot, and relaying the requisite feedback back to the framework through CCI. While CCI will be developed using the same programming language as the framework, CCM will be based on the cobot's operational language. Consequently, a standardized protocol should be established to facilitate data transfer between these modules, tailored to accommodate the specific requirements of both components.

IV. EXPERIMENTAL VALIDATION

The novel framework presented in the previous section underwent rigorous testing to validate its capabilities through a collaborative case study conducted within a defined collaborative work cell. This section comprehensively discusses the

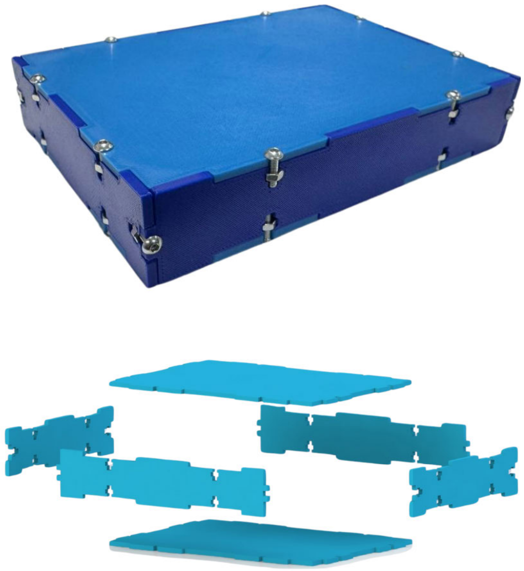


FIGURE 3. The box assembled during the collaborative task.

case study and workcell setup, implementation of CogniVera, experimental methodology, metrics utilized for evaluation, and corresponding results obtained from the experiments.

A. COLLABORATIVE TASK CASE STUDY

This case study examines the collaborative assembly of a box between a YuMi cobot and a human operator, drawing inspiration from Toichia et al [20]. As shown in figure 3, the box comprises six components assembled using screws and bolts. In the context of collaborative assembly processes, the cobot assumes a pivotal role by positioning components whereas the human operator engages in tightening screws and bolts.

The cobot operates according to the guidance provided by the operator and sequentially holds each component in place as shown in Figure 4. Vocal interactions facilitate communication between the cobot and the operator, ensuring effective collaboration throughout the assembly process.

B. COGNIVERA IMPLEMENTATION

For this case study, the Cognivera framework integrates advanced speech processing and control mechanisms to enable vocal interaction between the cobot and operator. The Speech Interface Module (SIM) leverages the OpenAI's Whisper module alongside their text-to-speech module to facilitate seamless communication between humans and cobots. The Cognitive Processing Unit (CPU) is built using OpenAI's "gpt-4o-2024-08-06" model [32], with both agents structurally prompted according to the established architecture and specifications to ensure optimal performance. Given the existing constraints of the model's context window, the experiments were conducted in six distinct phases, with five tasks sequentially completed within each phase. While this limitation poses challenges, future developments are anticipated to alleviate these issues.

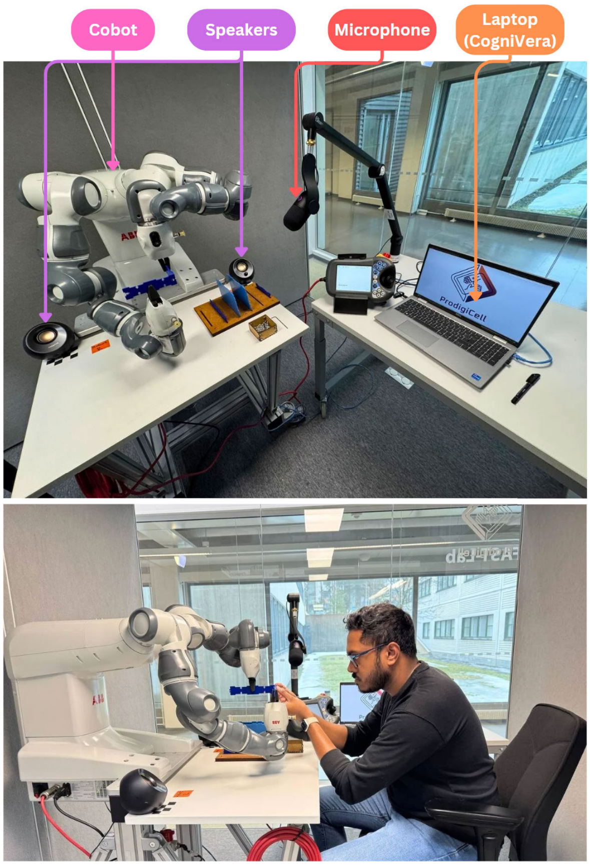


FIGURE 4. Collaborative task setup.

Through appropriate manipulation of the objects within the output structure of the LLM-based CPU, as shown in Listing 2, the system is equipped with the capability to execute a series of actions sequentially to satisfy user requests. For instance, when prompted to initiate a process, the robot is programmed to complete two steps sequentially before requiring further user intervention. This functionality enables the system to perform tasks in a structured sequence while subsequently reporting to the user, who can then address the necessary tasks, such as tightening screws. These advanced capabilities are facilitated by adhering to a well-defined prompting framework, defined in Section III.

The Cobot Controller (CC) is implemented using a Python-based Cobot Control Interface (CCI) and a RAPID-based Command Coordination Module (CCM) for the YuMi cobot. The communication between these components is managed by socket communication to enable efficient data exchange. The entire Cognivera system is deployed on a computer interfaced with audio capture devices, and the cobot within the designated collaborative cell, where specified collaborative tasks are carried out.

C. EXPERIMENTAL SETUP

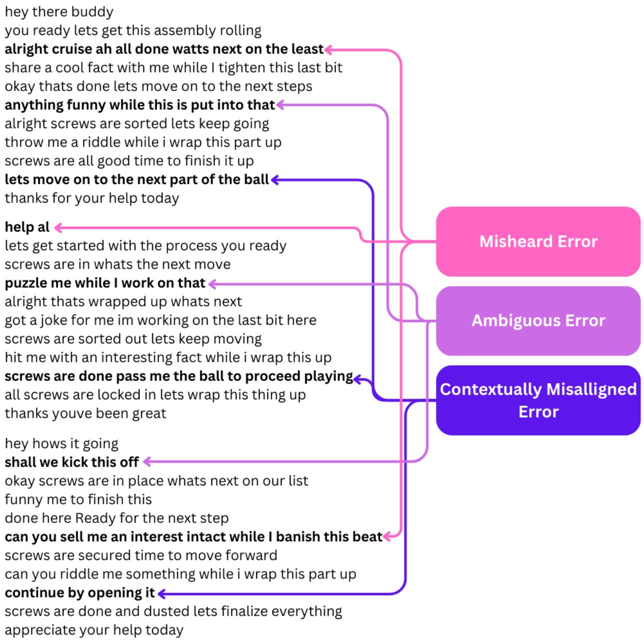
The defined collaborative task is executed using a cobot integrated with the established CogniVera framework within a collaborative work cell. This work cell was equipped with the necessary audio-receiving and emitting devices in conjunction with the cobot. Furthermore, it was soundproofed to maximise efficiency during task execution.

Initially, a skeleton script capable of performing a box assembly task was created as follows:

1) *Hello!* [Greeting]
2) *Start the assembly task* [Initiation of the process]
3) *I am done with the tightening. Can you do the next steps?* [Informing of human task completion and instructing to hold the next face (03) onto the workpiece]
4) *Tell me an interesting fact while I finish this up* [Requesting for entertainment (Interesting Fact)]
5) *Okay, I am done, let's move on to the next steps* [Informing of human task completion and instructing to hold the next face (04) onto the workpiece]
6) *Can you humor me with a joke* [Requesting for entertainment (Joke)]
7) *Done, let's move on* [Informing of human task completion and instructing to hold the next face (05) onto the workpiece]
8) *Ask me a Riddle* [Requesting for entertainment (Riddle)]
9) *Okay, let's finish this.* [Informing of human task completion and instructing to finish the work by placing it on the table]
10) *Thank You!* [End]

This script incorporated comprehensive instructions designed to guide the system during each phase of the task. This includes greetings and entertainment requests that enrich and validate the human-centric approach of the system. Moreover, these greetings function as separators between tasks, thereby facilitating the structured execution of continuous operations. Subsequently, 30 iterations of the skeleton were developed using the same GPT model as that used for the CRV process. In each iteration, three distinct types of errors were randomly introduced at various steps to validate error detection and correctness. These types of **errors** are as follows:

1) **Misheard Errors:** Errors that may arise when microphones do not accurately capture requests from the operator or when the speech-to-text system encounters difficulties in transcribing speech correctly. Such issues can occur due to confusion caused by similar-sounding words or misspellings, resulting in a misinterpretation of the request. Unfortunately, the language model may unintentionally treat this erroneous input as a legitimate instruction, which can generate unintended actions.
2) **Ambiguous Errors:** Ambiguous errors arise when a request is phrased in a manner that allows for multiple plausible interpretations, which makes it difficult for the system to determine the intended meaning with certainty. These errors typically occur because of inherent linguistic ambiguity, where different human operators may interpret the same request differently based on the context. Consequently, language models may misinterpret requests, leading to incorrect actions and inaccurate system outputs.



hey there buddy
you ready lets get this assembly rolling
alright cruise ah all done watts next on the least
share a cool fact with me while I tighten this last bit
okay thats done lets move on to the next steps
anything funny while this is put into that
alright screws are sorted lets keep going
throw me a riddle while i wrap this part up
screws are all good time to finish it up
lets move on to the next part of the ball
thanks for your help today

help al
lets get started with the process you ready
screws are in whats the next move
puzzle me while I work on that
alright thats wrapped up whats next
got a joke for me im working on the last bit here
screws are sorted out lets keep moving
hit me with an interesting fact while i wrap this up
screws are done pass me the ball to proceed playing
all screws are locked in lets wrap this thing up
thanks youve been great

hey hows it going
shall we kick this off
okay screws are in place whats next on our list
funny me to finish this
done here Ready for the next step
can you sell me an interest intact while I banish this beat
screws are secured time to move forward
can you riddle me something while i wrap this part up
continue by opening it
screws are done and dusted lets finalize everything
appreciate your help today

FIGURE 5. Examples of scripts included in the testing dataset with injected errors.

3) **Contextually Misaligned Errors:** These errors arise when a request fails to align contextually with the intended process. They may stem from human error or the heightened sensitivity of audio-capturing devices, which can inadvertently capture unrelated conversations. Furthermore, the hallucination tendencies characteristic of LLMs, as well as these erroneous requests, can result in outputs that prompt actions on the robot that are not in accordance with the collaborative task, thereby leading to potential failures.

In each iteration of the experiment, one error from each designated type was introduced, resulting in a total of three errors in randomly selected steps. Examples of some developed scripts are shown in Figure 5. This approach aims to facilitate a comprehensive observation of how the systems respond to these errors throughout the experimental process.

The **baseline** for this experiment, which serves as the point of comparison for the CogniVera framework, was defined as the same framework without the Validation agent. This configuration employs the Control Feedback loop described by Zhu et al. [7] and prompting techniques established in contemporary literature.

All data pertaining to the process will be systematically collected, including the requests submitted to the system, the output generated, and supplementary information such as accuracy scores and the time taken to process each request. The **metrics** employed to evaluate and analyze this system using the collected data throughout the experiment are outlined as follows:

1) **Error Detection:** This metric quantifies the rates of both the injected errors and common errors identified

TABLE 1. Overall performance metrics.

Metric	Process	Agent	Rate (%)
Error Detection	CRV	VA	70
		GA	28.89
		Total	98.89
	Conv		51.11
Error Rectification	CRV		97.78
	Conv		52.22
Task Completion	CRV		96.67
	Conv		40

by the system. It can be assessed using the accuracy score and output generated by the system.

2) **Error Rectification:** This metric defines the errors corrected by the system, wherein the identified errors are either accurately rectified during the generation of outputs or the user is solicited for further clarification, thereby preventing any unintended actions from the cobot.

3) **Task Completion:** This metric assesses the number of collaborative tasks that have been successfully executed. Success is characterized by the completion of all prescribed steps without any system failures resulting from unintended actions triggered by the system itself.

D. RESULTS AND DISCUSSION

The implemented test comprised 30 scripts and underwent analysis through two distinct methodologies: the CRV process, which incorporates the GA, and a conventional approach that excludes the GA. Five scripts were executed sequentially at a time, owing to the constraints of the context window of the language model. Results were systematically collected for this analysis. This methodology facilitated the execution of all 30 scripts across six distinct instances for both systems, thereby enabling comprehensive data collection and analysis. The generated data were collected and analyzed quantitatively in accordance with the established metrics (as presented in Table 1) and qualitatively to assess each request alongside its corresponding responses from the system.

Table 1 presents a comprehensive analysis of the overall performance metrics for the 30 collaborative tasks executed with injected errors, specifically comparing two defined scenarios: integration of the proposed CRV system and the traditional LLM-based system.

The results obtained from testing the proposed CRV-based system categorized the error detection outcomes into two distinct groups: the performance of the agents individually and the performance of the system as a whole. This methodological approach was aimed at validating the efficacy of incorporating VA into the system. The findings indicate that approximately 30% of the identified errors were detected exclusively by the GA rather than by VA. This observation does not suggest that the GA possesses an inherent capability to identify errors autonomously. Instead, it highlights the potential of LLMs in zero-shot learning, where the GA refines its error detection through continuous feedback received from the VA during the CRV process. Moreover, the results demonstrate that nearly 50% of the errors were undetected by the conventional system. In contrast, the system integrated with the CRV process failed to identify approximately 1% of the injected errors, totalling 90 errors. This substantial improvement underscores the efficacy of the CRV approach in enhancing error-detection capabilities.

The results for error rectification show that the CRV-integrated system achieved an error correction rate of nearly 98%. By contrast, the conventional system failed to rectify almost half of the introduced errors, demonstrating a significant performance improvement. This highlights the benefit of incorporating an additional validation layer to enhance the system's accuracy in handling errors.

In task execution, the CRV-integrated system successfully completed 29 out of 30 tasks, achieving an accuracy of 96.6%, whereas the conventional system only completed 12 tasks, with an accuracy of 40%. These results underscore the effectiveness and reliability of the proposed system in enabling accurate and robust LLM-based human-robot collaboration through vocal communication in industrial environments.

In Table 2, the performance metrics are analyzed in relation to the various types of errors introduced into the systems. This study seeks to observe the impact of these errors on the LLM-based system, particularly in relation to the emergence of hallucinations.

The findings indicate that the CRV-integrated system excels in error detection, effectively identifying all misheard and ambiguous errors while failing to detect a contextually misaligned error. This contextually misaligned error is the sole contributor to the system's unsuccessful task. In contrast, the conventional system exhibited moderate error detection proficiency, particularly in identifying misheard and ambiguous errors, with error rates surpassing 50%. However, it significantly underperforms in detecting contextually misaligned errors, achieving only a 33.33% detection rate.

Regarding error rectification, the CRV-integrated system demonstrated a high degree of efficacy and successfully rectified nearly all the errors. However, minor deficiencies exist in addressing ambiguous and contextually misaligned errors, both of which present a rectification rate of 96.7%. Conversely, the conventional system exhibits moderate success in rectifying misheard and ambiguous errors from the identified errors. However, it significantly falters in rectifying contextually misaligned errors, achieving an accuracy rate of merely 40%.

In summary, the CRV process reveals superior capabilities in detecting and correcting errors, particularly misheard and ambiguous errors. In comparison, the conventional process displays moderate detection capabilities but inferior

TABLE 2. **Performance metrics according to the type of error.**

Metric	Process	Agent	Misheard (%)	Ambiguous (%)	Contextual (%)
Error Detection	CRV	VA	100	43.3	66.67
		GA	0	56.67	30
		Total	100	100	96.67
	Conv		66.67	53.33	33.33
Error Rectification	CRV		100	96.67	96.67
	Conv		66.67	50	40

rectification rates, particularly for contextually misaligned errors. The results obtained from this experiment were thoroughly observed, leading to the identification of several significant features of the newly developed system. One of the principal observations was that the proposed system demonstrated the capability of identifying errors before they were executed and rectified, thereby facilitating a seamless flow of accurate actions in response to user requests. Furthermore, it was noted that the system effectively manages continuous tasks, during which hallucinations are most likely to occur, to complete nearly all tasks in only a single instance of failure.

The average time taken to process a single request in the proposed system was 5.47 seconds, compared to 3.26 seconds for the conventional method. Efficiency is best defined as the optimal combination of both accuracy and speed, rather than prioritizing one over the other. With the implementation of an internal language model trained explicitly for this purpose and sufficient computing power to support it, the processing time of the proposed system is expected to decrease significantly. These enhancements are anticipated to be the future improvements of this study.

The only failure of a collaborative task by the proposed system was identified as stemming from the injection of a contextually misaligned error. In this instance, when the user requested the system to "okay finish the process and give me the locks," the system erroneously progressed to the subsequent steps despite the absence of any locks to the process. This misalignment can be attributed to the ambiguity introduced by the phrase "finish the process," which led both the GA and the VA to mistakenly interpret the user's intent as a desire to complete the task. Consequently, both agents executed actions based on this presumption without adequately verifying the relationship between the request for locks and the directive to conclude the process. Therefore, future enhancements will concentrate on developing methodologies for training the LLMs to actively identify and rectify contextually misaligned errors.

Furthermore, the accuracy score obtained from the prompting rules was calculated through a statistical approach to enhance clarity. This assessment was conducted using the results derived from 90 injected synthetic errors. It can be denoted as follows:

$$P(\text{score} = 10 \mid \text{Accurate Response}) = 28.9\% \tag{2}$$

$$P(\text{score} \in [0, 9] \mid \text{Accurate Response}) = 0\% \tag{3}$$

$$P(\text{score} = 10 \mid \text{Failed Response}) = 1.11\% \tag{4}$$

$$P(\text{score} \in [6, 9] \mid \text{Failed Response}) = 0\% \tag{5}$$

$$P(\text{score} = 5 \mid \text{Failed Response}) = 25.56\% \tag{6}$$

$$P(\text{score} = 4 \mid \text{Failed Response}) = 0\% \tag{7}$$

$$P(\text{score} = 3 \mid \text{Failed Response}) = 2.22\% \tag{8}$$

$$P(\text{score} \in [1, 2] \mid \text{Failed Response}) = 0\% \tag{9}$$

$$P(\text{score} = 0 \mid \text{Failed Response}) = 42.22\% \tag{10}$$

The diverse scores attributed to the failed responses reflect the responses identified as needing minor revisions due to small inaccuracies. This statistical analysis enhances our understanding of how the VA functioned to safeguard the accuracy of responses and the overall actions of the system.

In summary, the results validate the proposed CRV process as a robust and scalable approach for integrating LLMs into collaborative robotic systems, marking a significant advancement toward the development of more reliable and human-centered HRC solutions.

V. CONCLUSION

This study presents a novel framework that integrates LLM-based agents to enhance HRC through vocal interaction in industrial environments. By introducing a dual-agent architecture, the system effectively mitigates hallucinations and improves reliability, thereby surpassing conventional single-agent approaches. The proposed CRV process plays a crucial role in refining error detection and rectification, as demonstrated by the significant reduction in undetected errors and improvement in task execution accuracy.

Experimental validation using a structured assembly task confirmed the superiority of the CRV-integrated system. The findings indicate a remarkable improvement in error detection and rectification, with the system achieving a 96.6% task completion rate compared to the conventional approach's 40% success rate of the conventional approach. In addition, the system's ability to refine its performance through continuous feedback highlights the potential of

LLMs in zero-shot learning for industrial HRC. These results underscore the effectiveness of the proposed framework for enabling robust, reliable, and human-centric cobot interactions. Future research will investigate the deployment of the proposed methodologies within more complex industrial contexts. This will involve engaging the general public in experiments aimed at assessing, understanding, and further advancing the existing system.

REFERENCES

[1] Directorate-General for Research and Innovation (European Commission), M. Breque, L. De Nul, and A. Petridis. (2021). *Industry 5.0: Towards a Sustainable, Human Centric and Resilient European Industry. Publications Office of the European Union*. [Online]. Available: https://data.europa.eu/doi/10.2777/308407

[2] T. Wang, J. Fan, and P. Zheng, "An LLM-based vision and language cobot navigation approach for human-centric smart manufacturing," *J. Manuf. Syst.*, vol. 75, pp. 299–305, Aug. 2024, doi: 10.1016/j.jmsy.2024.04.020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0278612524000864

[3] A. Bauer, D. Wollherr, and M. Buss, "Human–robot collaboration: A survey," *Int. J. Humanoid Robot.*, vol. 5, no. 1, pp. 47–66, Mar. 2008, doi: 10.1142/s0219843608001303. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/S0219843608001303

[4] Y. Kim, D. Kim, J. Choi, J. Park, N. Oh, and D. Park, "A survey on integration of large language models with intelligent robots," 2024, *arXiv:2404.09228*.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*.

[6] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," 2022, *arXiv:2209.07753*.

[7] J. Ye Zhu, C. Gomez Cano, D. Vazquez Bermudez, and M. Drozdzal, "InCoRo: In-context learning for robotics control with feedback loops," 2024, *arXiv:2402.05188*.

[8] B. Xie, X. Xi, X. Zhao, Y. Wang, W. Song, J. Gu, and S. Zhu, "ChatGPT for robotics: A new approach to human–robot interaction and task planning," in *Intelligent Robotics and Applications*, H. Yang, H. Liu, J. Zou, Z. Yin, L. Liu, G. Yang, X. Ouyang, and Z. Wang, Eds., Cham, Switzerland: Springer, pp. 365–376, doi: 10.1007/978-981-99-6495-6_31.

[9] C. Wang, S. Hasler, D. Tanneberg, F. Ocker, F. Joublin, A. Ceravola, J. Deigmoeller, and M. Gienger, "LaMI: Large language models for multi-modal human–robot interaction," in *Proc. Extended Abstr. CHI Conf. Human Factors Comput. Syst.*, May 2024, pp. 1–10, doi: 10.1145/3613905.3651029.

[10] P. Allgeuer, H. Ali, and S. Wermter, "When robots get chatty: Grounding multimodal human–robot conversation andcollaboration," in *Artificial Neural Networks and Machine Learning*, M. Wand, K. Malinovsk, J. Schmidhuber, and I. V. Tetko, Eds., Cham, Switzerland: Springer, pp. 306–321, doi: 10.1007/978-3-031-72341-4_21.

[11] A. Urlana, C. Vinayak Kumar, A. Kumar Singh, B. Mallikarjunarao Garlapati, S. Rao Chalamala, and R. Mishra, "LLMs with industrial lens: Deciphering the challenges and prospects - a survey," 2024, *arXiv:2402.14558*.

[12] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," *ACM Comput. Surveys*, vol. 55, no. 12, pp. 1–38, Dec. 2023. [Online]. Available: https://dl.acm.org/doi/10.1145/3571730

[13] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, "Smart factory of industry 4.0: Key technologies, application case, and challenges," *IEEE Access*, vol. 6, pp. 6505–6519, 2018, doi: 10.1109/ACCESS.2017.2783682. [Online]. Available: https://ieeexplore.ieee.org/document/8207346

[14] S. Robla-Gómez, V. M. Becerra, J. R. Llata, E. González-Sarabia, C. Torre-Ferrero, and J. Pérez-Oria, "Working together: A review on safe human–robot collaboration in industrial environments," *IEEE Access*, vol. 5, pp. 26754–26773, 2017, doi: 10.1109/ACCESS.2017.2773127. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8107677

[15] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," *Auto. Robots*, vol. 42, no. 5, pp. 957–975, Jun. 2018, doi: 10.1007/s10514-017-9677-2.

[16] J. De Schutter, T. De Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbeliën, K. Claes, and H. Bruyninckx, "Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty," *Int. J. Robot. Res.*, vol. 26, no. 5, pp. 433–455, May 2007, doi: 10.1177/027836490707809107.

[17] D. J. Agravante, A. Cherubini, A. Bussy, P. Gergondet, and A. Kheddar, "Collaborative human-humanoid carrying using vision and haptic sensing," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 607–612. [Online]. Available: https://ieeexplore.ieee.org/document/6906917

[18] O. M. Al-Jarrah and Y. F. Zheng, "Arm-manipulator coordination for load sharing using reflexive motion control," in *Proc. Int. Conf. Robot. Autom.*, vol. 3, 1997, pp. 2326–2331. [Online]. Available: https://ieeexplore.ieee.org/document/619309

[19] L. Rozo, S. Calinon, D. Caldwell, P. Jimenez, and C. Torras, "Learning collaborative impedance-based robot behaviors," in *Proc. AAAI Conf. Artif. Intell.*, 2013, vol. 27, no. 1, pp. 1422–1428. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/8543

[20] A. T. Eyam, W. M. Mohammed, and J. L. M. Lastra, "Emotion-driven analysis and control of human–robot interactions in collaborative applications," *Sensors*, vol. 21, no. 14, p. 4626, Jan. 2021, doi: 10.3390/s21144626. [Online]. Available: https://www.mdpi.com/1424-8220/21/14/4626

[21] A. Cherubini, R. Passama, A. Crosnier, A. Lasnier, and P. Fraisse, "Collaborative manufacturing with physical human–robot interaction," *Robot. Comput.-Integr. Manuf.*, vol. 40, pp. 1–13, Aug. 2016, doi: 10.1016/j.rcim.2015.12.007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736584515301769

[22] J. T. C. Tan, F. Duan, Y. Zhang, K. Watanabe, R. Kato, and T. Arai, "Human–robot collaboration in cellular manufacturing: Design and development," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 29–34. [Online]. Available: https://ieeexplore.ieee.org/document/5354155

[23] S. Hjorth and D. Chrysostomou, "Human–robot collaboration in industrial environments: A literature review on non-destructive disassembly," *Robot. Comput.-Integr. Manuf.*, vol. 73, Feb. 2022, Art. no. 102208, doi: 10.1016/j.rcim.2021.102208. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736584521000910

[24] T. Wang, P. Zheng, S. Li, and L. Wang, "Multimodal human–robot interaction for human–centric smart manufacturing: A survey," *Adv. Intell. Syst.*, vol. 6, no. 3, Mar. 2024, Art. no. 2300359, doi: 10.1002/aisy.202300359. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/aisy.202300359

[25] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, W. Ye, Y. Zhang, Y. Chang, P. S. Yu, Q. Yang, and X. Xie, "A survey on evaluation of large language models," *ACM Trans. Intell. Syst. Technol.*, vol. 15, no. 3, pp. 31–45, 2024. [Online]. Available: https://dl.acm.org/doi/10.1145/3641289

[26] A. Radford and K. Narasimhan. (2018). *Improving Language Understanding By Generative Pre-training*. [Online]. Available: https://www.semanticscholar.org/paper/Improving-Language-Understanding-by-Generative-Radford-Narasimhan/cd18800a0fe0b668a1cc19f2ec95b5003d0a5035

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[28] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "LLaMA: Open and efficient foundation language models," 2023, *arXiv:2302.13971*.

[29] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. Singh Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. Le Scao, T. Lavril, T. Wang, T. Lacroix, and W. El Sayed, "Mistral 7B," 2023, *arXiv:2310.06825*.

[30] *Meet Claude anthropic*. Accessed: Feb. 7, 2025. [Online]. Available: https://www.anthropic.com/claude

[31] DeepSeek-AI et al., "DeepSeek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning," 2025, *arXiv:2501.12948*.

[32] OpenAI et al., "GPT-4 technical report," 2023, *arXiv:2303.08774*.

[33] T. B. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2020, pp. 1877–1901. [Online]. Available: https://papers.nips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

[34] P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," 2017, *arXiv:1706.03741*.

[35] H. Liu, Y. Zhu, K. Kato, A. Tsukahara, I. Kondo, T. Aoyama, and Y. Hasegawa, "Enhancing the LLM-based robot manipulation through human–robot collaboration," *IEEE Robot. Autom. Lett.*, vol. 9, no. 8, pp. 6904–6911, Aug. 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10561501

[36] S. Izquierdo-Badiola, G. Canal, C. Rizzo, and G. Alenyà, "PlanCollabNL: Leveraging large language models for adaptive plan generation in human–robot collaboration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2024, pp. 17344–17350. [Online]. Available: https://ieeexplore.ieee.org/document/10610055

[37] L. Xia, Y. Hu, J. Pang, X. Zhang, and C. Liu, "Leveraging large language models to empower Bayesian networks for reliable human–robot collaborative disassembly sequence planning in remanufacturing," *IEEE Trans. Ind. Informat.*, vol. 21, no. 4, pp. 3117–3126, Apr. 2025, doi: 10.1109/TII.2024.3523551. [Online]. Available: https://ieeexplore.ieee.org/document/10834394

[38] F. Gao, L. Xia, J. Zhang, S. Liu, L. Wang, and R. X. Gao, "Integrating large language model for natural language-based instruction toward robust human–robot collaboration," *Proc. CIRP*, vol. 130, pp. 313–318, Jan. 2024, doi: 10.1016/j.procir.2024.10.093. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827124012502

[39] A. Brohan et al., "RT-2: Vision-Language-Action models transfer Web knowledge to robotic control," 2023, *arXiv:2307.15818*.

[40] Z. Mandi, S. Jain, and S. Song, "RoCo: Dialectic multi-robot collaboration with large language models," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2024, pp. 286–299. [Online]. Available: https://ieeexplore.ieee.org/document/10610855

[41] C. Gkournelos, C. Konstantinou, and S. Makris, "An LLM-based approach for enabling seamless human–robot collaboration in assembly," *CIRP Ann.*, vol. 73, no. 1, pp. 9–12, 2024, doi: 10.1016/j.cirp.2024.04.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S000785062400012X

[42] M. U. Farooq, G. Kang, J. Seo, J. Bae, S. Kang, and Y. J. Jang, "DAIM-HRI: A new human–robot integration technology for industries," in *Proc. IEEE Int. Conf. Adv. Robot. Social Impacts (ARSO)*, May 2024, pp. 7–12, doi: 10.1109/ARSO60199.2024.10557811. [Online]. Available: https://ieeexplore.ieee.org/document/10557811

[43] D. Zhang, M. Van, S. Mcllvanna, Y. Sun, and S. McLoone, "Adaptive safety-critical control with uncertainty estimation for human–robot collaboration," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 4, pp. 5983–5996, Apr. 2023, doi: 10.1109/ARSO60199.2024.10557811. [Online]. Available: https://ieeexplore.ieee.org/document/10281398

[44] A. Brohan et al., "RT-1: Robotics transformer for real-world control at scale," 2022, *arXiv:2212.06817*.

[45] Z. Ji, T. Yu, Y. Xu, N. Lee, E. Ishii, and P. Fung, "Towards mitigating hallucination in large language models via self-reflection," 2023, *arXiv:2310.06271*.

[46] C. Severance, "Discovering Javascript object notation," *Computer*, vol. 45, no. 4, pp. 6–8, Apr. 2012, doi: 10.1109/MC.2012.132. [Online]. Available: https://ieeexplore.ieee.org/document/6178118/citations#citations

NADUN RANASINGHE (Graduate Student Member, IEEE) received the B.Eng. degree (Hons.) in electrical and electronics engineering from Curtin University, Perth, WA, Australia, in 2020, and the M.Sc. degree in factory automation and robotics from Tampere University, Finland, in 2024, where he is currently pursuing the Ph.D. degree in engineering sciences, funded by the Doctoral School of Artificial Intelligence, hosted by the Finnish Center for Artificial Intelligence. Since 2022, he has been a part of the FAST-Laboratory Research Group, contributing to European projects, such as AI-PRISM and AIDEAS, focusing on the integration of AI-based systems in industrial robotics. He is a Ph.D. Researcher with Tampere University. His current research explores the integration of advanced AI models, such as large language models (LLMs), with robots to enhance human-centric interaction in industrial settings. His research interests include collaborative robotics, factory automation, machine learning, and large language models.

WAEL M. MOHAMMED received the B.Sc. degree in mechatronics engineering from The University of Jordan, in 2010, the M.Sc. degree in automation engineering from Tampere University of Technology, in 2017, and the Ph.D. degree in engineering sciences from Tampere University, in 2024. He is currently a Postdoctoral Research Fellow with Tampere University. He is the Technical Manager of the FORTIS project. Additionally, he has been involved in writing research and innovation proposals funded by the European Commission. In 2011, he was the Head of the Technical Department on the Traffic Management System project at Etihad Alafandi L.L.C. in Saudi Arabia. In 2015, he joined FAST-Laboratory and has been working there since. His research interests include robotics, digital twins, human–robot interaction, knowledge-based reasoning engines, and factory automation.

KOSTAS STEFANIDIS received the Ph.D. degree in personalized data management from the University of Ioannina, Greece. He is currently a Professor of data science with the Faculty of Information Technology and Communication Sciences, Tampere University, Finland, where he also leads the Data Science Research Centre and the Group on Recommender Systems. He has more than ten years of experience in different roles at ICS-FORTH, Greece, NTNU, Norway, and CUHK, Hong Kong. His work focuses on personalization and recommender systems, entity resolution, data exploration, and data analytics, with a special focus on socio-technical aspects in data management, such as fairness and transparency, and has been published in more than 100 papers in top-tier conferences and journals. He has been involved in several international and national research projects and is also actively serving the scientific community. His research interest includes the broader area of big data. He is the General Co-Chair of the ADBIS 2025, TPDL 2025, and EDBT/ICDT 2026.

JOSE L. MARTINEZ LASTRA (Member, IEEE) received the Ingeniero Tecnico Industrial degree in electrical engineering from the Universidad de Cantabria, Santander, Spain, and the M.Sc. and Dr.Tech. degrees (Hons.) in automation engineering from Tampere University of Technology, Tampere, Finland. He joined Tampere University of Technology, in 1999, and became a Full Professor of factory automation, in 2006. Previously, he carried out research with Departamento de Ingenieria Electrica y Energetica, Santander; the Institute of Hydraulics and Automation, Tampere; and the Mechatronics Research Laboratory, Massachusetts Institute of Technology, Cambridge. He has published more than 250 original papers published in international journals and conference proceedings. His research interests include the application of information and communication technologies in the fields of factory automation and robotics. He is a member of the IEEE Industrial Electronics Society and several editorial boards and has served as an Associate Editor for IEEE Transactions on Industrial Informatics, in 2006 and from 2012 to 2022, and the Technical Editor for IEEE/ASME Transactions on Mechatronics, from 2015 to 2016. He was the Deputy Chair of the IEEE/IES Technical Committee on Industrial Cyberphysical Systems, from 2019 to 2022.