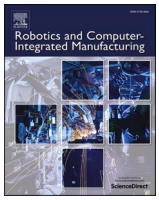
Robotics and Computer–Integrated Manufacturing 97 (2026) 103120

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim



LLM based autonomous agent of human-robot collaboration for aerospace wire harnessing assembly

Yiwei Wang [a,b], Qi Guo [a], Lianyu Zheng [a,b,*], Binbin Wang [a], Pai Zheng [c], Zhonghua Qi [a,d]

[a] School of Mechanical Engineering and Automation, Beihang University, Beijing 100191, China
[b] MIIT Key Laboratory of Intelligent Manufacturing Technology for Aeronautics Advanced Equipment, Ministry of Industry and Information Technology, Beijing 100191, China
[c] Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, China
[d] 29th Research Institute of China Electronics Technology Group Corporation, China

ARTICLE INFO

Keywords:
Large language model agent
Human-robot collaboration
Wire harnessing assembly

ABSTRACT

The fusion of large language models (LLMs) and robotic system bring transformative potential to human-robot collaboration (HRC). Existing LLMs-based HRC methods mainly realize on fine-tune techniques, which has the shortcomings such as damage of the inherent ability of original LLMs, difficulty performing complex continuous task, less flexibility, fixed response strategy and computationally expensive. Alternatively, the development paradigm of LLM applications is transiting towards the autonomous agent mode. This paper proposed an interesting LLM agent based HRC framework (or HRC agent), which empowers the robot with human's think mode and execution ability of sensing, interaction, self-reasoning, task planning and task execution. The chain-of-thought technique that generates a series of intermediate reasoning steps is adopted to improve the ability of LLMs to execute complex reasoning and task. Few-shot learning is used such that HRC agent can quickly learns new specific industry tasks by being provided a few examples. The reflection-based contextual memory mechanism enables HRC agent to have long term memory and continuous instruction understanding ability. A series of tools are developed and integrated into HRC agent, by which the capabilities of HRC agent can be easily expanded without much changing of the code framework. The functionality and effectiveness of HRC agent is validated in the aerospace wire harnessing assembly task, whose products has the characteristics of small diameter wires, complicated wire text, dense and tiny assembly holes, varying product batch size and customized production, and thus has high requirements for flexibility. The results show that the HRC agent is able to well understand the natural language instructions and give correct and effective response by chain-of-though, and subsequently, drive the robot to execute tasks correctly by calling tools.

1. Introduction

The increasingly deep integration of industrial Artificial Intelligence (AI) technology [1] and manufacturing provides more flexible and intelligent operating paradigm for industrial robots in manufacturing sectors such as machining, assembly, and additive manufacturing. As an emerging flexible manufacturing paradigm, human-robot collaboration (HRC) fully combines the flexibility, proficiency, and knowledge intensity of humans with the high speed, high load, high repeatability, and high positioning accuracy of industrial robots [2,3]. The advantages of both human and collaborative robot complementarity provide new solutions for complex assembly tasks with high efficiency and precision. Major manufacturing countries and research institutions are gradually forming a consensus on the importance of human-machine collaboration [4]. Human-robot collaboration has become one of the key factors to achieve intelligent upgrading of the manufacturing industry.

Currently, most existing robotic systems conduct the pre-programmed tasks in a routine manner with poor interaction ability and limited intelligences, let along the well-handle personalized tasks in a collaborative approach can be promising. The planner for robot execution and human operations is normally predefined, lacking real-time adjustment capabilities during task fulfilment progress. The complexity in programming theses robots remains a significant barrier, open requiring specialized knowledge and hindering the ease of modification and re-programming [5].

The trend of varying batch size, customized production in

* Corresponding author.
E-mail address: lyzheng@buaa.edu.cn (L. Zheng).

https://doi.org/10.1016/j.rcim.2025.103120
Received 23 December 2024; Received in revised form 19 June 2025; Accepted 28 August 2025
Available online 2 September 2025

manufacturing industry requires robots having advanced abilities of sensing, interaction, cognizing, reasoning, decision-making, execution, learning, and consequently, handling task-change more flexible. In this context, the robotics programming has transitioned from rigid coding, i.e., one task one programming to more adaptable, human-centric mode. The fusion of large-scale generative models, such as Large Language Models (LLMs), Visual-Language Models (VLMs), multi-modal models, etc., and robotic systems has introduced a transformative paradigm in HRC [6], which enables the robot to not only simply execute human instructions but also to understand the semantics contained in the voice instructions and moreover, to achieve semantics reasoning. This allows the operators to control the robot through natural language in a more natural, straightforward, and programming-free way, enabling the robot to deal-with task-change more flexibly.

The reminder parts of this paper are organized as follows. Section 2 reviews related works regarding HRC and large-scale generative models empowered robots. The research gaps are summarized, followed by proposing the motivation and contribution of this paper. Section 3 presented the overview of the proposed framework.

2. Related works

2.1. Human-robot collaborative assembly

Currently, the research regarding HRC mainly focuses on the following aspects including robot control, safety, task allocation, collaborative time prediction, context-aware, human intention recognition, augmented reality (AR) assisted HRC. The following literature review focuses on the above aspects. For robot control, Andronas et al. [7] proposed a robotic system involving detection, grasping, cross section recognition and assembly to address the manufacturing process automation of linear non-rigid components. Monguzzi et al. [8] proposed a hierarchical optimal model-based path planning strategy for robotic manipulation of deformable linear objects. Liu et al. [9] investigated multimodal data-driven robot control for human-robot collaborative assembly. Zhao et al. [10] proposed an HRC framework for robotic assembly based on impedance control, in which, four control modes including position control, drag and drop, positive impedance control and negative impedance control were developed for the robot. Focusing on safety, Zhang et al. [11] proposed a human-cyber-physical assembly system framework that combines the perception and control capacity of digital twin with the virtual-reality interaction capacity of augmented reality (AR) to achieve a safe and efficient HRC environment. Aiming at task allocation, Krishna et al. [12] proposed assembly line digital twin and collaborative robot work simulation tools to estimate the robot's assembly task performing ability and execution time, which were used to optimize the work allocation between human and robot. Evangelou et al. [13] introduced a solution in scheduling and allocation of assembly tasks to both human and robot for effective emergence of alternative task-resource assignment sequences. Li et al. [14] developed a temporal subgraph reasoning-based method for self-organizing HRC task planning between multiple agents. Banziger et al. [15] introduced an approach to use standardized work description for automated procedure generation of mobile assistant robots in HRC to optimize human-robot task allocation. For finding the right collaborative timing for robotics in HRC and to avoid inappropriate collaborative behaviors, Zhang et al. [16] proposed a fusion-based spiking neural networks for collaboration request prediction, in addition, they proposed a reinforcement learning algorithm to optimize the task sequence allocation scheme in assembly processes [17]. Adaptable and natural interaction is an important goal yet to be achieved in HRC. Context-aware, intention recognition, perception and reasoning are support technologies for cognitive HRC. Zheng et al. [18] proposed a visual reasoning-based approach for mutual-cognitive HRC to fully integrate the robotic and human cognitions. Male et al. [19] proposed a cognitive architecture composed of perception and reasoning modules that allows a robot to adapt its actions while collaborating with humans in assembly task. Zhang et al. [20] proposed a human-object integrated approach for context-aware assembly intention recognition in HRC to better recognize the operator's intention. Li et al. [21] developed a multimodal transfer-learning-enabled action prediction method to predict the operator's intentions in advance. For potential collision detection in HRC applications, Makris et al. [22] proposed a vision system. Augmented Reality (AR) is increasingly introduced in HRC to enhance the mutual cognition between human and robot. Li et al. [23] used AR to handle the multi-robot teleoperation tasks. The robots were mapped into the AR glasses and three interactive AR-assisted modes including real-time motion control, planned motion control and robot monitoring mode were generated. Chu et al. [24] designed an interface in AR for HRC assembly and verified the effectiveness of visual and haptic cues that convey the robot intent to human. And in our previous work, we also proposed a set of human-centered large-scale deployable mechanism collaborative assembly system based on AR equipment, and improved the level of human-machine collaboration system through three digital twins.

2.2. Large-scale generative models empowered human-robot collaboration

Large-scale generative models including large language models (LLMs), Visual-language models (VLMs), and multi-modal models are models with abilities to generate text [25], image [26], video [27], code [28], etc. The fusion of these models with robots enables a more flexible HRC, offering unparalleled capabilities that facilitate the process of task specification, allowing robots to receive, interpret, understand natural language and execute the high-level instructions from human. In early studies, Large-scale generative models were commonly employed as text processing tools for parsing robotic commands [29]. In recent years, with the emergence of multi-modal models such as PaLM-E [30], RT-2 [31], and VoxPoser [32], the combination of large-scale generative models with robots has unveiled a novel research direction.

For robot interaction, some research provides textual environment descriptions [33,34] or integrates Visual Language Models (VLMs) with LLMs [35,36]. An impressive work is from Huang et al. [33], which combines LLMs and VLMs to map language instructions into 3D value maps to guide robot trajectories. Wu et al. [37] developed a personalized robot assistant empowered by LLMs for the household cleanup scenario. The robot can tidy up rooms by picking up objects and putting them away, during which the robot learned the user preferences and reapplied this preference in its future cleaning up tasks. Obinata et al. [38] used the large-scale VLM to detect semantic changes in daily life environment for household tasks.

Within the industrial scenes such as manufacturing and assembly, Gkournelos et al. [39] introduced an LLM-based manufacturing execution system that enabled the system to interact with robots in digital twin environment with natural language. Aristeidou et al. [40] presented a framework using generative AI to implement a dynamically updateable perception system to enhance autonomy and applicability of robot across diverse scenarios. Zheng et al. [41] integrated vision-language cues with LLM and proposed a vision-language-guided HRC task planning approach. Park et al. [3] proposed a framework to allow human to interact with robots using natural language instructions for pick-and-place construction operations. Li et al. [42] developed a natural language-enabled virtual assistant named Max to support flexible and scalable human–robot interactions (HRI) with industrial robots. Yin et al. [43] fine-tuned LLMs to translate human language instructions into reward functions and guide a deep reinforcement learning module to generate robot executable actions. Although LLMs have demonstrated their great potential to accomplish a wide range of tasks in the form of question-answering (QA), building autonomous agents is far from QA since they need to fulfill specific roles and autonomously perceive and learn from the environment to evolve themselves like humans. Currently, the development paradigm of LLM applications is transiting

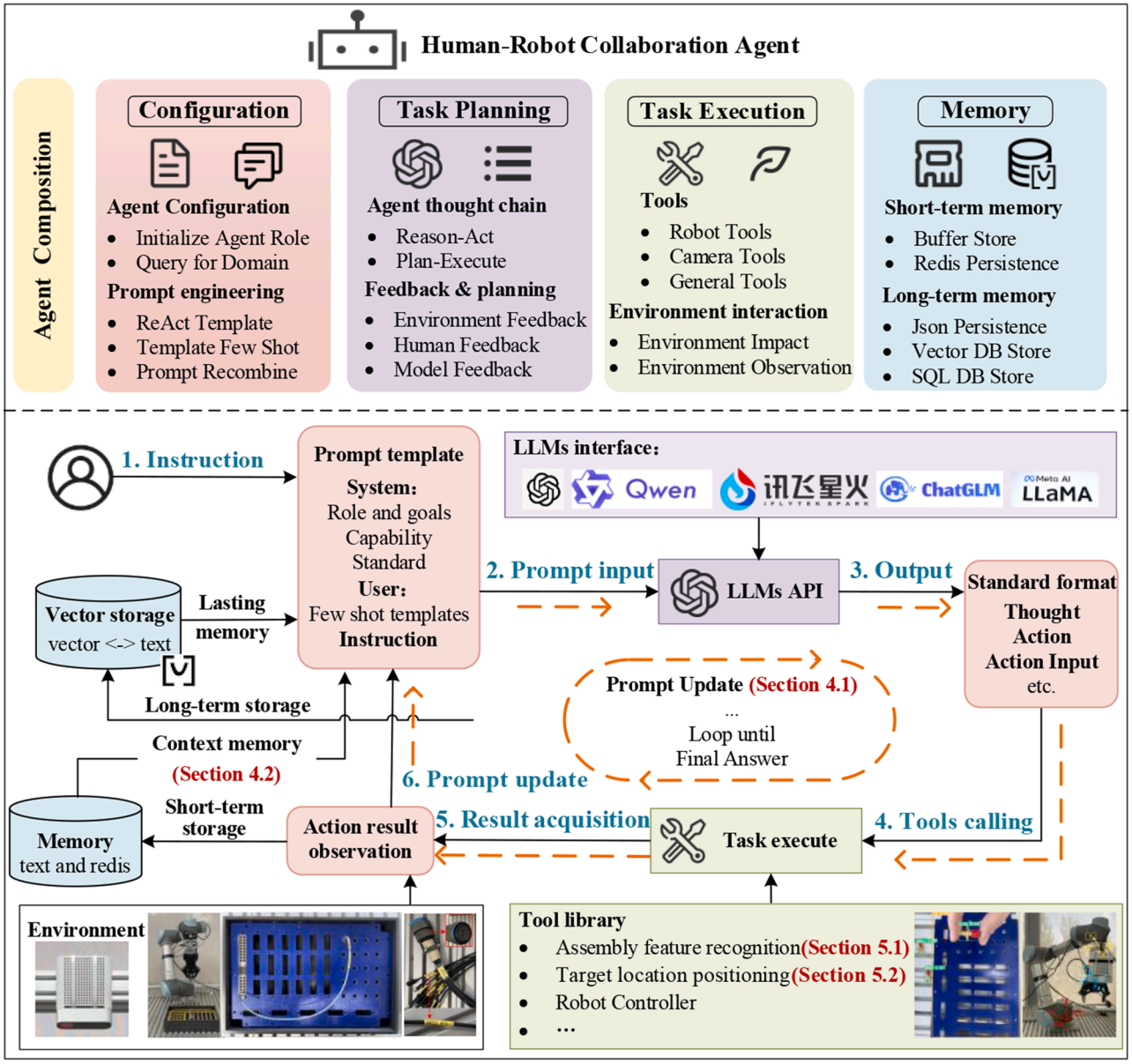


Fig. 1. The composition and working mechanism of HRC agent.

towards the AI Agent mode. Although the topic of combining LLMs with robot has many publications, designing an LLM-agent is far more than using LLMs on robot, which is much more flexible and scalable than purely using LLMs. In the embodied AI area, the publication that uses the LLM-agent framework on the robot and realizes a complete industry case is still very few. Fan et al. [44] delves into the potential of LLM agents for industrial robotics with an emphasis on autonomous design, decision-making and task execution with manufacturing contexts. Kannan et al. [45] proposed an embodied multi-robot task planning framework for home scene. The above work is either in simulation environment or in the general scenarios rather than industry applications.

In addition, it is necessary to briefly discuss the traditional robot planning and execution methods (rule-based, symbolic logic-based and reinforcement learning-based). Rule-based task planning methods describes the task flow through predefined state transition diagrams, finite state machines (FSMs) or behavior trees. They have the advantages such as simple implementation, fast response, and suitable for structured environments while with shortcomings such as lack of flexibility, difficulty in adapting to dynamic environments. Symbolic logic-based planning methods first describes the task using language such as PDDL (Planning Domain Definition Language) and then call a planner to generate executable sequences. They can handle complex logical reasoning tasks but rely on predefined rules and logic and thus difficult to handle fuzzy or unstructured information. Reinforcement learning-based methods use algorithms such as Deep Q-Network (DQN) and Proximal Policy Optimization (PPO) to train robots to complete specific tasks. They can learn optimal strategies in complex environments. But the training process is time-consuming and the strategy is poorly interpretable.

2.3. Research gap and motivation

Based on the above literature reviews, the research gaps and the motivations are summarized below. The traditional LLM mechanism is based on question-answering form, which requires the LLM to complete a task continuously. But since the model only has short-term memory or no memory, it is difficult for the LLM model to autonomous perception

and interacting with environment, reflection and revision, and thus evolution. The development paradigm of LLM applications is transiting towards the AI Agent mode. Comparing with the traditional QA mechanism of LLM, the AI agent mechanism is a dynamic and iterative process. The agent is first given a complex task and then it will formulate a work plan and decompose the rough task into a series of sub-task. Here the complex tasks refer to those that cannot be completed by with only a single tool calling and thus need to be decomposed to a series sub-task until meta-task. Meta-tasks refer to tasks that can be completed by a single tool calling or by calculations of the LLM itself. During the execution of the sub-tasks, agent will memorize, evaluate and self-correct its action. Under this mechanism, AI Agent does not only react to inputs, but also can proactively think, plan and make decisions to complete complex tasks like an autonomous agent. Therefore, this mechanism enables AI agent to optimize its work through iterations, and in addition, to solve unplanned tasks.

In summary, towards the post-LLM era, aiming at reformulating the LLM application development with the idea of autonomous agent, and consequently expecting to obtain a more flexible and intelligent HRC system, this paper proposes a new HRC framework. The contributions are as bellow.

(1) An autonomous agent mechanism-based HRC framework (abbreviated as HRC agent) is proposed. In the agent, the original LLMs without fine-tuned serve as the "brain" of the agent to understand the high-level natural language instructions from human, and plan and decompose the complex task into meta-tasks. A series models or algorithms (such as object detection model, coordinate transformation algorithm) are trained or developed, encapsulating as "tools" of the agent. The industrial robot serves as the "execution" part of the agent to final complete tasks. In summary, the agent is with abilities of parsing natural language instruction, understanding ambiguous and fuzzy expression, sensing, cognizing, planning and decomposing tasks, execution, interacting with environment and self-correction. HRC agent is a highly modular framework that is very easy to transfer or customize to other applications. Once the framework is built up, one can expand the capabilities of the agent by adding as many customized or general tools as needed without changing much the whole code framework.

(2) In the proposed HRC agent, both the specialized and general tools are integrated. The specialized tools include the trained high-precision object detect models, the coordinate transformation algorithms etc., which are customized developed for the specific industrial application such as wire harnessing assembly. The general tools include some ready-made models that are used for general tasks such as context awareness, audio processing, etc. This enables the proposed HRC Agent to not only adapt to general scenarios, but also to high-precision industrial applications, being compatible to both general and specialized scenes.

(3) Comparing to the fixed QA workflow or routine of the traditional LLM + robot solution, the "chain-of-thought prompting" ability of HRC agent ameliorates the flexibility and robustness of the HRC system, which (a) enable the system to self-reasoning complex task when given an natural language instructions, (b) enables the system to deal with unplanned task, (c) give rationale reaction to some exceptional errors, and (d) improves the fault tolerance ability during task execution. In addition, the few-shot learning and context memory mechanism enables HRC agent to have the continuous conversation ability. The reflection-summary mechanism is used to address continuous conversation-induced token overload problem.

The proposed HRC method is used and validated in a high-precision assembly scenario of wire harnessing in aviation electronic equipment. The product has the characteristics of varying batch size, dense and small size assembly features, which requires high execution accuracy for robot. To our best knowledge, this is one of the first works that combines the cutting-edge AI agent idea with LLM, and further uses this agent to solve a high-precision industrial problem.

3. Methodology of proposed HRC agent

3.1. Constituent components of HRC agent

To empower the HRC agent with human's think mode and execution ability, four modules, i.e., configuration module, task planning module, task execution module, and memory module, are designed and integrated into the HRC agent, as shown in Fig. 1. The ***configuration*** module defines the characteristics and behaviors of the HRC agent through a series of parameters and rules, describing the attributes of the agent such as its role, goal, ability, knowledge and behavior. These attributes determine how the agent will think, understand, and plan the tasks, and subsequently, response and execute the task while interact with environment simultaneously. ***Task planning*** module decomposes complex tasks into sub-task and develops corresponding execution strategies. There are two thinking modes named Reason-Act mode [46] and Plan-Execute during task planning. For Reason-Act mode, the results of previous action are fed back to the agent to re-plan the next action before executing new actions. For Plan-Execute mode, the agent first decomposes the complex into sub-tasks and then executes them by Reason-Act mode. In both the two modes, the agent continuously obtains feedbacks from environment and users during its thinking and planning, which ensures the output actions are reasonable, i.e., the actions are continuous and executable. In addition, vision system is used to capture the environment information and feedback to HRC agent. ***Task execution*** module transforms planning to executing, which connects the internal decision of the agent and external environment. This is realized by customizing a series of "tools" and teaching the agent how to use these tools. The ***memory*** module stores the information obtained from environment and users during interaction, which will be used to guide the future actions. In terms of structures, memory consists of short-term and long-term memory with the former one temporarily stores recent perceptions in order to assist in guiding the planning of the current task, while the latter one stores historical execution records in a persistent manner to guide the planning of all related tasks. In terms of format, memory can be expressed in natural language or encoded as vectors to improve retrieval efficiency. In terms of operation, memory interacts with environment through reading, writing and reflections, in which reflection summarizes and compress the information to save persistent storage space.

In summary, original LLM without fine-tuned serves as the "brain" of the HRC agent and a series of customized tools are developed as its "hands" to execute tasks. The vision system serves as the "eyes" of HRC agent to capture the environment information. Functional logic and working mechanism of HRC agent is designed such that its "brain", "eye", "hands" can work in reasonable coordination.

3.2. Working mechanism of HRC agent

To ensure the modules of the HRC agent operates in a stable, orderly and reasonable manner, a complete working mechanism of agent is designed, as shown in Fig. 1, which includes six procedures, i.e., (1) user issues instruction, (2) prompt construction, (3) LLM output, (4) tools configuration and calling, (5) execution result acquisition, (6) prompt update and reorganization.

The user first issues an input "instruction" to the agent. The "instructions" can be in various forms such as text and voice. Voice command will be converted into text through automatic speech recognition (ASR). The text-formatted instruction is concatenated into the existing prompt words to form the initial prompt (The whole process of constructing the initial prompt words will be detailed later). The above functions of are mainly implemented by the "Configuration" module of the agent.

Then the agent enters the planning and execution phase, which will be repeated in cycle until the HRC task is completed. In the cycle, the following fives procedures are implemented. The initial prompt is firstly

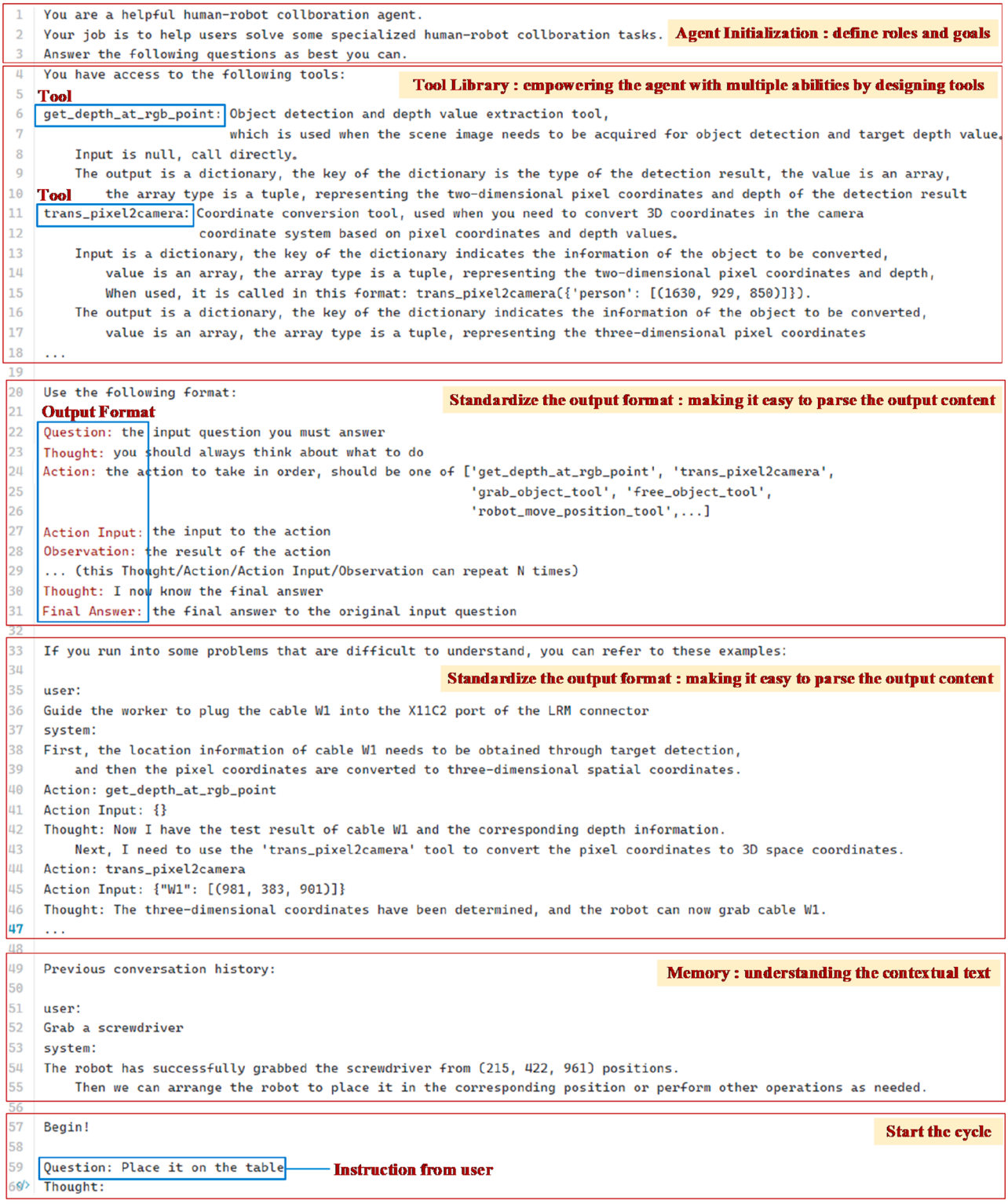


Fig. 2. Structure of the initial prompt.

fed into the LLM, which can be any original LLM without fine-tuned such as ChatGPT, ChatGLM, etc. Since the specification of the output format is defined in the prompt template, the output of the LLM is also in an easy-to-parse format. The content of the LLM output includes three components denoted as "Thought", "Action" and "Action input", which completely describe the task that need to be executed by the agent. Among these, "Thought" describe the purpose of current procedure and the tools that may be used. "Action" is the name of the tools, and "Action input" is the arguments accepted by the tool. The above functions of are mainly implemented by the "Task Planning" module of the agent.

After parsing the content of the LLM output, the tools in the tool library are matched and the arguments is passed through the tool calling chain to execute the actions. The execution results or the impact to the environment of this action execution will be observed by the agent and recorded in a component denoted as "Observation". The above functions of are mainly implemented by the "Task Execution" module of the agent.

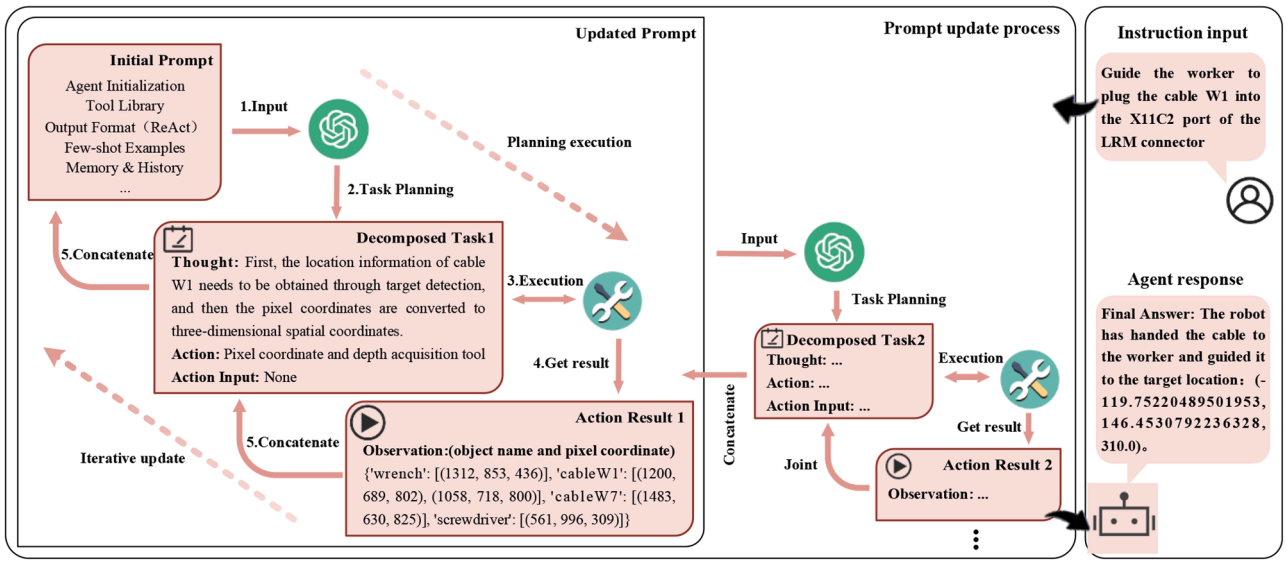


Fig. 3. The update mechanism of prompt.

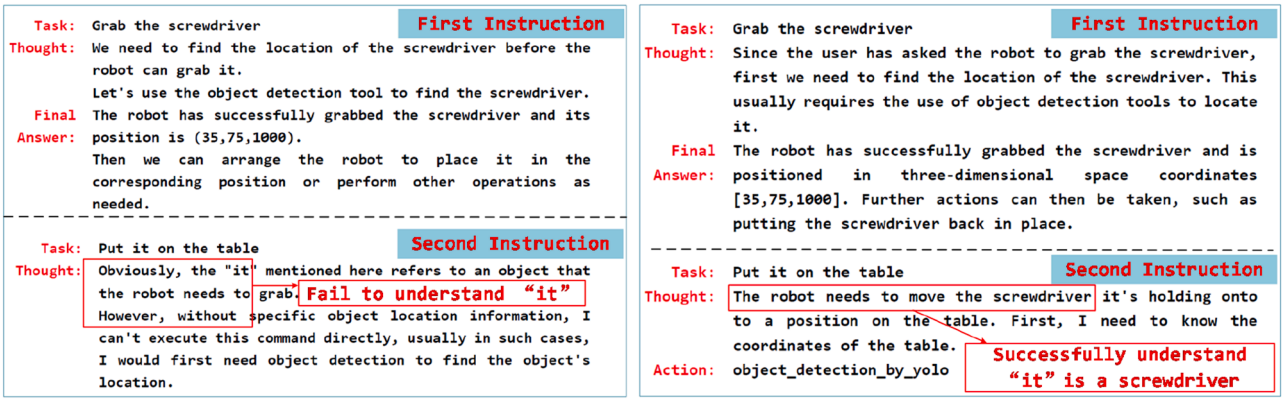


Fig. 4. Comparison of continuous conversation with and without contextual memory.

Finally, before a new cycle to execute the next action, the "Thought", "Action", "Action Input" and "Observation" of the current cycle are concatenated into the initial prompt word in the form of text or vector to form a new prompt. Therefore, in addition to the agent configuration information and the user input, the new prompt also contains the historical execution records, which will be used as the "experiences" to guide the agent in its following task planning and execution. During the working of the agent, the short-term operation records will be stored in the form of "Memory", which provides contextual memory such that the agent can achieve continuous conversation with human and thus understand human contextual instructions. While the long-term operation records will be vectored and stored in a vector database through the reflection-summary mechanism to achieve long-term memory persistence, which facilitates the retrieval of the agent and improve the efficiency and stability of the agent. The above functions of are mainly implemented by the "Memory" module of the agent.

4. Prompt template design of HRC agent

4.1. Initialization and update mechanism of prompt

The prompt construction, including prompt initialization and prompt update, is essential to the functionalities of the agent. The structure of the initial prompt is shown in Fig. 2, where the tool library, few-shot learning, memory and the user input are generated in real-time during the running process while the other texts are the fixed template. In the prompt, the description of tools establishes the connection between the "Configuration" module and "Task execution" module of the agent. The few-shot learning mechanism provides examples of execution process of some specific complex tasks in professional industry scenarios such that the "Task Planning" module (or LLM) of the agent can rapidly learn how to generate solution for similar tasks. In this case, when the agent encounters similar task in specific domain with strong professionalism, it can generate feasible solutions according to user demand.

In the initial prompt template, standardizing the output format is important, where the output specification of the LLM after understanding the task is defined. The Reason-Act [46] chain of thought framework is adopted. Note that "Reason-Act" and "Plan-Execute" are two main "Chain-of-thought" modes in current agent systems. "Reason-Act" mode is a "think while doing" way of working while "Plan-Execute" makes a complete plan at the beginning and then implement it step by step. "Reason-Act" has better real-time response performance and adaptability to dynamic environments and thus more suitable for unstructured or uncertain scenarios. "Reason-Act" mode is used in this paper since the HRC assembly scenario is highly flexible and uncertain.

The chain of thought framework allows the LLM to generate reasoning trajectories and planning steps in an intertwined manner. The generated reasoning trajectories enable LLM to induce, track and update

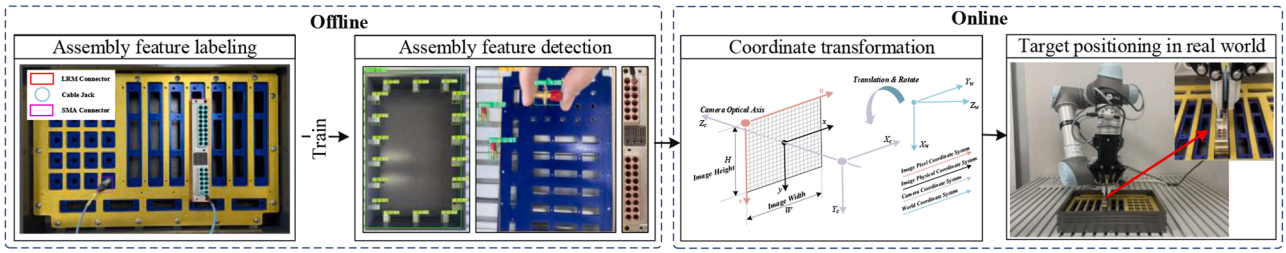


Fig. 5. Process of assembly feature visual understanding and target location positioning.

Table 1
mean Average Precision for YoloV8 in wire assembly scenario.

Parts	Sample size	AP	mAP@0.5	mAP@0.5:0.95
chassis	645	94.2 %	95.4 %	68.3 %
bracket	1632	92.0 %		
SMA	1401	97.6 %		
LRM	1367	95.1 %		
cable	1506	98.1 %		
screwdriver	398	95.5 %		

the planning steps and handle abnormal situations. During the planning steps, the agent interacts with external resources such as database and environment. The core process is "thinking→action→observation→output", which is defined in the prompt template. Then the LLM will generate output that contains the three components "Thought", "Action" and "Action input" according to the user's instructions, as illustrated in Fig. 2. After the execution of actions, the "Observation" along with the above three components are concatenated to update the prompt. The above process of "prompt update" is illustrated in Fig. 3**.** This "chain-of-thought" mode is particular benefit for executing complex tasks is composed of several simple tasks.

4.2. Continuous instruction understanding based on contextual memory

In order to make the HRC agent operates in a more human-like mode and improve the interactivity, the contextual memory is introduced into the HRC agent to realize human-robot continuous conversation. Currently, there are two ways to call the LLMs, i.e., call them by API interface or call them through dialogue on the Web page. The former is often unable to store memory for continuous conversation due to the limitations of API calling ways. In contrast, the latter adds a historical conversation content to each input when developing chatbot applications, which realizes contextual continuous conversation. According to Section 4.1, each time the HRC agent receives a task, it will enter the chain-of-thought to perform the task planning and execution cycle until the task is completed and a "Final Answer" (see Fig. 2) is generated, which records the final execution result of a task. This "Final Answer" is concatenated to the prompt when the HRC agent plan a new instruction. Note that this mechanism is different from the prompt update in Section 4.1. The prompt update occurs during the chain-of-thought inside the instruction execution process while the recording and concatenating of "Final Answer" happens when the user issues a new instruction.

Fig. 4 compares the continuous conversation with and without contextual memory. In the scenario without contextual memory, the agent is first told to "grab the screwdriver" and it is able to correctly understand and execute the instruction. However, as the user continues to issue instruction to the agent such as "Put it on the platform", the agent will confuse and its response is unclear. Due to lack of contextual memory, the agent is not able to understand what "it" in the second instruction refers to. In contrast, in the scenario with contextual memory, when provided with the same two instructions, the agent can well understand and even put forward the requirements for completing the task, showing strong ability of self-thinking. Furthermore, when provided the third instruction, based on the context, the agent can well understand the whole task and execute correctly.

However, the above contextual memory based continuous conversation also brings new problem. As the number of consecutive task executions increases, the number of tokens input into the prompt words will also increase even exceed the limit, which in turn may truncate the prompt words and decrease the reasoning speed. The reflection-summary mechanism is used to address this problem. When the number of historical records reaches a preset threshold, an existing LLM interface (such as Qwen2.5) is called to summarize multiple historical records and generate a concise new record. An example is taken below. When the agent completes three tasks and stored the following three records in the memory module:

- The continuous dialogue robot with no context memory has successfully grabbed a screwdriver and is positioned in three-dimensional space coordinates [35,75,1000]. Further actions can then be taken, such as tightening the screws or putting the screwdriver back in place
- The robot has put the screwdriver on the table
- The robot has moved to its initial home position and is ready for further operations

By calling the LLM interface, the system summarizes the above three records as: "The robot successfully grabbed the screwdriver and placed it on the table, and when the task was complete, the robot returned to its initial position and could continue operating." This effectively reduces the storage for redundant historical information and thus controls the number of tokens while keep the benefits of contextual memory, consequently, improves the reasoning efficiency of the system.

5. Assembly feature visual understanding and target location positioning

One benefit of using HRC agent is that once the framework is built up, one can expand the capabilities of the agent by adding as many customized or general tools as needed without changing much the whole code framework. The assembly feature visual understanding is executed by the improved object detection model YoloV8 while target location positioning is executed by the customized coordinate transformation algorithm. All of these two model and algorithm are encapsulated as "tools" and are called by HRC agent when needed. In addition, other general tools are developed and integrated inside the HRC agent. The tool list is given in Table 5 in Appendix and only the tools of visual understanding and target location positioning are detailed below.

5.1. Assembly feature recognition

After "Task Planning", the "Task execution" needs to be implemented by robot. For manual assembly such as wire harnessing assembly of electronic equipment, the robot first needs to detect the target holes according to the assembly process and then to accurately position the hole to guide the operator. Wire harnessing assembly has the nature of dense and small size assembly features. Here the deep learning model is

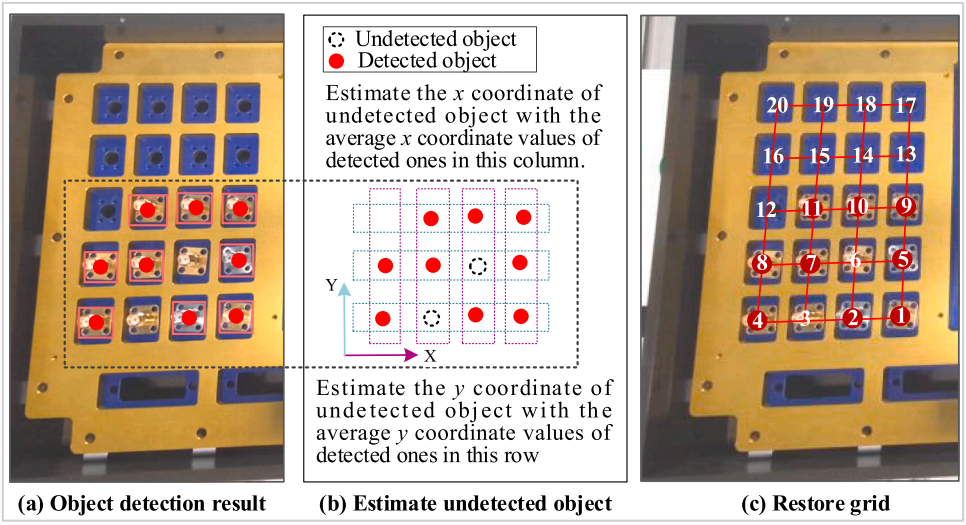


Fig. 6. Grid mapping method for undetected objects.

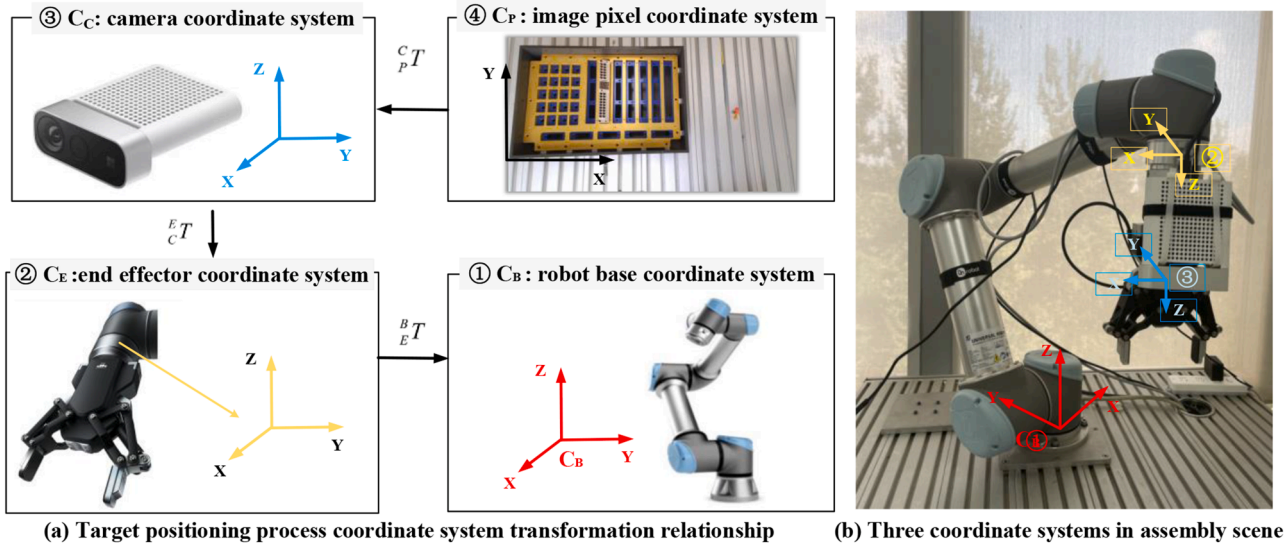


Fig. 7. Transformation between different coordinate systems.

used for object detection. The whole process of assembly feature visual understanding and target location positioning includes online and offline stages, as shown in Fig. 5. The offline stage deals with the assembly features detection. 300 photos are taken with different angles and all the assembly features are labeled. The original photos are then reinforced using different types of reinforcement methods such as rotating, mirroring, adding noise, blurring, stitching. Finally, 788 photos are obtained. Then the labeled photos are used to train the object detection model. Here the State-of-The-Art (SOTA) model of one-stage object detection YoloV8 is used. After training, the mean Average Precision (mAP@0.5) is around 95.4 %, as shown in Table 1.

For majority objects in wire assembly scenario, the trained YoloV8 model can successfully detect and obtain their pixel coordinates. However, for the assembly array composed of dense and tiny feature, due to the variation of light and camera angle, the trained YoloV8 model may not achieve a 100 % detection accuracy. For example, as shown in Fig. 6 (a), nine out of eleven SMA connectors are successfully detected while two of them are failed to be detect. In such case, the standard vision pipeline is not enough. In practice, feature array such as the "holes array" is a kind of very common assembly feature in aerospace products, and solving the precise detection of such feature array is very important and challenging. Therefore, a "grid matching method" based on the assembly feature distribution is presented as a complementary to cover all the installation holes, as shown in the Fig. 6(b). For an undetected object, its x pixel coordinate value is estimated by the average of x coordinate values of detected ones in the column it is in while its y pixel coordinate value is estimated by the average values of y coordinates of detected ones in the row it is in.

Since it is known in advance that the SMA connector are distributed in a grid shape, the pixel coordinates of undetected SMA connectors can be estimated by the above method, by which way, the grid can be restored in pixel coordinate system. Then the assembly features are numbered from the lower right corner of the grid and then are assigned a unique pixel coordinate value, by which, the mapping of SMA connector from physical word position to the pixel coordinate system is achieved.

5.2. *Converting pixel coordinate to world coordinate*

After the target assembly feature is detected on image, the spatial positioning of the target involves the coordinates transformation between different coordinate systems, which are listed below and illustrated in Fig. 7. Note that to ensure the completeness of the content, the

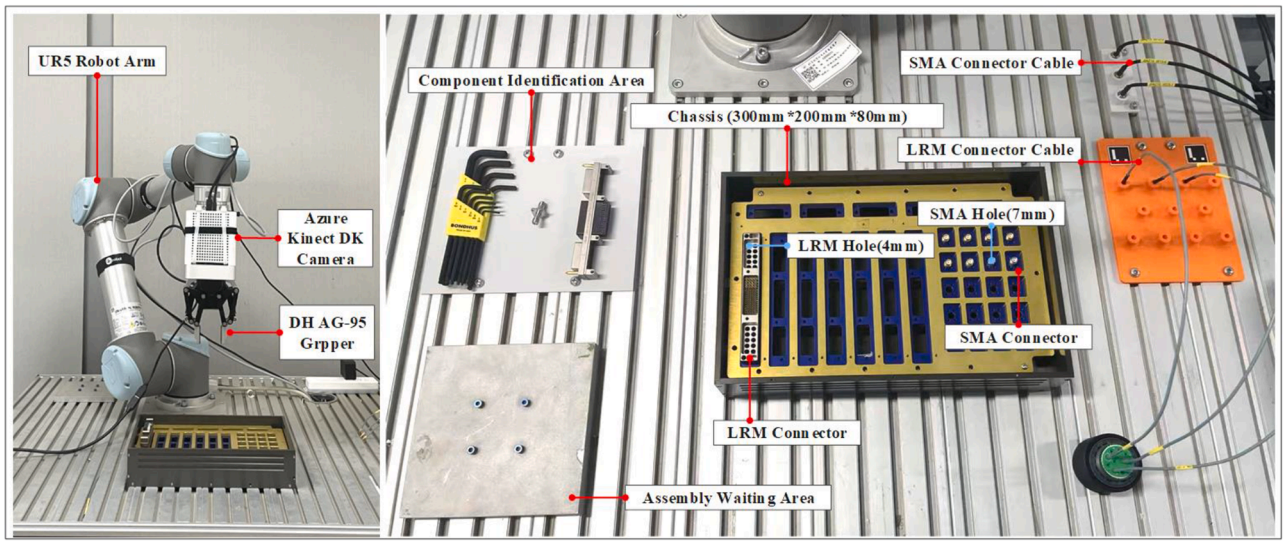


Fig. 8. HRC Agent system composition.

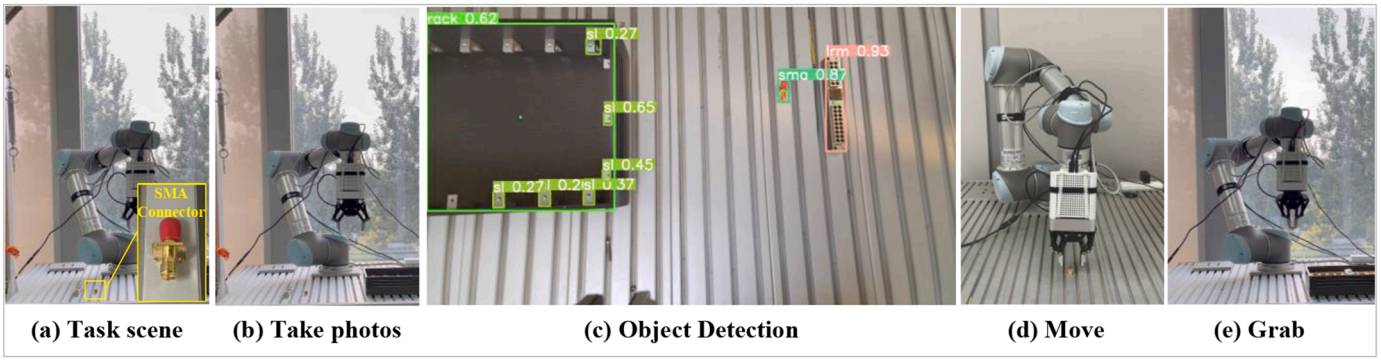


Fig. 9. Executing the task of pick up the SMA connector.

Chain-of-think automatically generated by HRC agent

Thought: Depth image and RGB image have been captured, now need to perform SMA target detection on RGB image.
Action: object_detection_by_yolo
Action Input: {"rgb_image": "image_path"}
Observation: The target detection has been completed, and the pixel coordinates of each detection result in the image are as follows:
The pixel coordinates of sma are: (801, 419) ① Object detection to get pixel coordinate on image
Thought: SMA has been detected on the RGB image, and its depth information needs to be obtained next.
Action: get_object_depth
Action Input: {"cls": "sma", "pixel_coordinates": [[801, 419]], "image_path": "image_path"}
Observation: Depth value of sma at point (801, 419) on rgb image: 345 mm ② Object depth
Thought: The depth information of the SMA has been obtained and can now be translated into the camera coordinate system.
Action: trans_pixel2camera
③ Object target position on the camera coordinate system
Action Input: {"cls": "sma", "pixel_coordinates" : [[801, 419, 345]]}
Observation: On the RGB image, the coordinates of point sma in the camera coordinate system are: (-61.3, -48.8, 345.0)
Thought: The coordinates of the SMA in the camera coordinate system are known, and the next step is to convert them to the arm base coordinate system.
Action: trans_camera2base
④ Object target position on the base coordinate system
Action Input: {"cls":"sma", "pixel coordinate": [-61.3, -48.8, 345.0]}
Observation: The coordinates of sma in the frame coordinate system of the manipulator are: [-0.255, -0.357, 0.028]

Fig. 10. Partial "chain-of-though" results for instruction "pick up the SMA connector".

Table 2
Tasks designed to validate the "target recognition success rate" (TRR).

Group 1		
Objective: To validate the ability of HRC agent to understand instructions as well as the ability of detection model to recognize the target. **Criteria for task successful**: Recognize the targets and complete the grabbing/placing tasks correctly.		
No. Task	**Instruction**	**TRR**
1.1	Get a screwdriver	10/10
1.2	Pick up the screwdriver	10/10
1.3	Grab the bracket	10/10
1.4	Pick up the bracket	10/10
1.5	Grab a SMA connector	9/10
1.6	Pick up the SMA connector	10/10
1.7	Get the LRM connector	10/10
1.8	Pick up the LRM connector	10/10
1.9	Put the SMA connector into the rack	10/10
1.10	Put the LRM connector into the rack	10/10
Group2		
Objective: To validate the grid matching method for building the assembly array graph and consequently, the ability of HRC agent to understand the graph. Assembly array refers to the array that comprises same type of assembly features such as ports, SMA connectors, LRM connectors, etc. **Criteria for task successful**: Identify and then move to the target port correctly. Identify and then grab the target wire correctly.		
No. Task	**Instruction**	**TRR**
1.11	Move to the port of SMA1 connector	10/10
1.12	Move to the port of SMA5 connector	10/10
1.13	Move to the port of SMA10 connector	10/10
1.14	Grab cable w1	9/10
1.15	Grab cable w3	10/10
1.16	Grab cable w5	8/10
1.17	Grab cable w6	10/10

basic theory of coordinate transformation is briefly presented.

Robot base coordinate system C_B: The original point of C_B is located at the center of the base plan. Its XOY plane coincides with the base surface and Z axis is perpendicular to the based surface and upward. End gripper coordinate system C_E: This coordinate system is set up through the UR robot controller. Based on the C_B, the original point of C_E is defined as the center point of the end gripper in the fully opened state. Camera coordinate system C_C: The camera's own coordinate system. The camera is fixed on the end tool of the robot. The target position directly obtained by vision method is the coordinates in this coordinate systems. Image pixel coordinate system C_P: a two-dimensional coordinate system based on image captured by depth camera. The coordinates of the point in the image are pixel values.

The process of converting the object detection results (which is the pixel coordinates of the target assembly feature) into the target position in the real world will undergo three coordinate transformations. Given that the pixel coordinates of the target assembly feature is$p_p = (u,v)$, it is easily to transform$p_p = (u,v)$to the camera coordinate system C_C and obtain the coordinates of point in C_C, denoted as $p_C = (x_C, y_C, z_C)$. The Tsai-Lenz algorithm and a 15 × 16 chessboard are used to perform the hand-eye calibration to obtain the coordinate transformation matrix between C_C and C_E, which is denoted as${}^{E}_{C}T$and is shown in Eq.(1) after homogeneous processing.

$$ {}^{E}_{C}T = \begin{bmatrix} {}^{E}_{C}R & {}^{E}_{C}Tv \\ 0 & 1 \end{bmatrix} \tag{1} $$

where${}^{E}_{C}R$and${}^{E}_{C}Tv$are rotation matrix and translation vector, respectively. Then the point$p_C = (x_C, y_C, z_C)$ in the C_C can be transformed into a point$p_E = (x_E, y_E, z_E)$in the end gripper coordinate system C_E by$p_E = {}^{E}_{C}T \bullet p_C$.

C_E is set up through the robot controller, after which, the pose of the end gripper in the robot base coordinate system C_B is obtained, denoted as$({}^{B}_{E}Tv, {}^{B}_{E}\theta)$. ${}^{B}_{E}Tv$is the translation vector from C_B to C_E while ${}^{B}_{E}\theta$is the internal rotation Euler angel of the C_E in C_B. Then the rotation matrix from C_B to C_E can be obtained, denoted as${}^{B}_{E}R$. After homogeneous processing, the coordinate transformation matrix between C_E and C_B is shown in Eq.(2).

$$ {}^{B}_{E}T = \begin{bmatrix} {}^{B}_{E}R & {}^{B}_{E}Tv \\ 0 & 1 \end{bmatrix} \tag{2} $$

Then the point$p_E = (X_E, Y_E, Z_E, 1)$in the end gripper coordinate system C_E is transformed into the point$p_B = (X_B, Y_B, Z_B, 1)$in the robot base coordinate system C_B by$p_B = T \bullet p_E$. Finally, the coordinates of point P_B is fed into the robot to drive the robot to move to P_B.

After being integrated in the HRC agent, the "tools" control the robot arm, the end-gripper, depth camera and other hardware in ROS (Robot Operating System). The version is ROS2 Humble. For example, the end-gripper is controlled in the ROS based on the Modbus TCP protocol. The driver of UR5 robot comes from "Universal_Robots_ROS2_Driver" and the depth camera driver comes from "Azure_Kinect_ROS2_Driver". The project is developed using multiple languages including Python, *C++*, etc. The entire project contains a total of 543 files of more than ten file types (such as Python, *C++*, CMake, XML, YAML, etc.). The functionalities are encapsulated as "ROS function package". After enabling the hardware driver in the ROS system, control of each hardware device is implemented by the "tool" through "topics" and "services" mechanism of ROS.

6. Case study: wire harnessing assembly guidance with the HRC agent system

Assembly occupies an important position in the aerospace manufacturing industry, accounting for a large proportion in terms of labor force, material resources, and working hours. In modern manufacturing, assembly time accounts for 40 % to 60 % of the entire manufacturing time [47]. Compared with highly automated industries such as the automotive industry, aerospace products are characterized by customization, variable batch sizes, large numbers of components, and complex processes, leading to low automation. Therefore, the aerospace products are still mainly manually assembled. The low degree of automation and intelligence of the current manual assembly mode makes it difficult to meet the demands for multi-model, large-batch, and short-cycle customization of aerospace products.

Wire harnessing is one of the most significant parts of electronic equipment in industries such as aerospace, shipbuilding, automobile, etc. Currently, the wire harnessing assembly has the following difficulties. 1) The assembly is pure manually operation that the operators need to insert the wire into the right hole according to the wiring process. 2) The assembly features such as wires and holes are dense, tiny and diverse. The operators need to frequently switch between these assembly features and the wiring process diagram, resulting in heavy cognitive load, high error and omission rates. The above problems lead to low first-time pass rate and long assembly cycle of electronic equipment assembly, which is difficult meet its high-efficiency and high-precision production needs.

The proposed HRC agent system is applied on wire harnessing assembly to address the above problem, as shown in Fig. 8. The collaborative robot used in this paper is a UR5 robot developed by Universal Robots Corporation. The UR5 robot has a payload of 5 kg, a working radius of 850 mm , and a repeatability of ±0.03 mm , which makes it suitable for precision operation. The robot weighs 18.4 kg, which is convenient for flexible installation in the aerospace industry scenario. The robot has built-in force sensing and collision detection such that human-robot collaboration is safe without a safety fence.

Fig. 8(b) shows a typical wire harnessing assembly scenes of aerospace industry. The assembly features are the "dense tiny holes" in the "assembly object", which is a chassis used in electronic equipment of aerospace. The assembly task is to insert the wires into these holes. There are two types of wires and correspondingly two types of holes, i.e.,

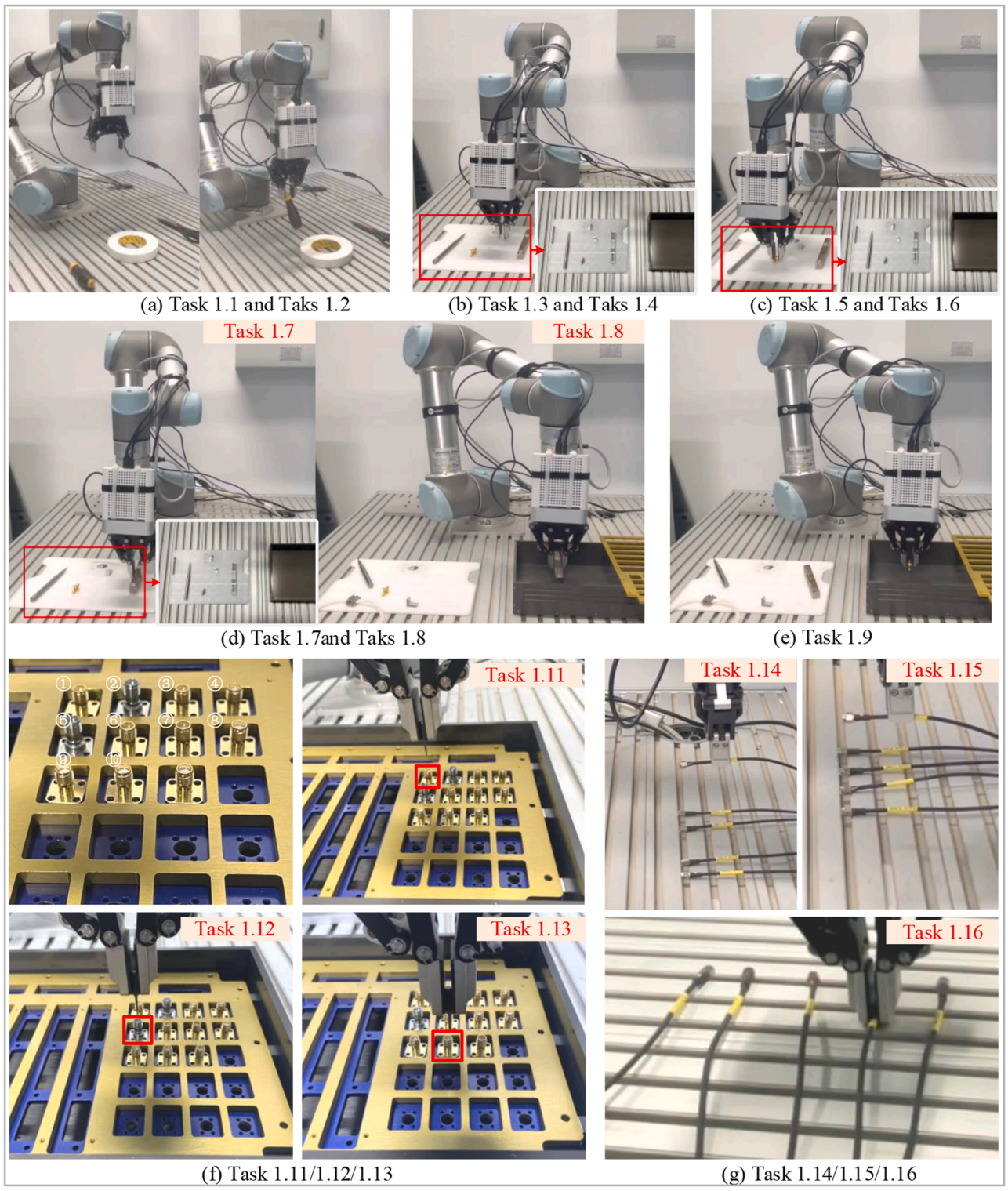


Fig. 11. Verification of Task 1.1-Task 1.16.

the holes on LRM connector (abbreviated as LRM holes) and the holes for installing SMA connector (abbreviated as SMA holes). The diameter of LRM hole is 4 mm and the SMA hole is 7 mm . The diameters of the wires to be installed in these two types of holes are even smaller. Therefore, it is a very challenging task. Take LRM connector as an example, there are dozens of tiny holes with diameters of 4 mm on it, it is a very labor intensive with high cognitive load task, leading to low efficiency and quality.

The abilities of self-sensing, self-planning and self-executing are the three core capabilities of the HRC agent. Therefore, the HRC agent is validated progressively from three levels: (1) Target recognition ability, which is quantifies by the metric of target recognition success rate (TRR), (2) Task planning ability, assessing by the metric of task planning success rate (TPR) and (3) Task execution ability, assessing by the metric of task execution success rate (TER).

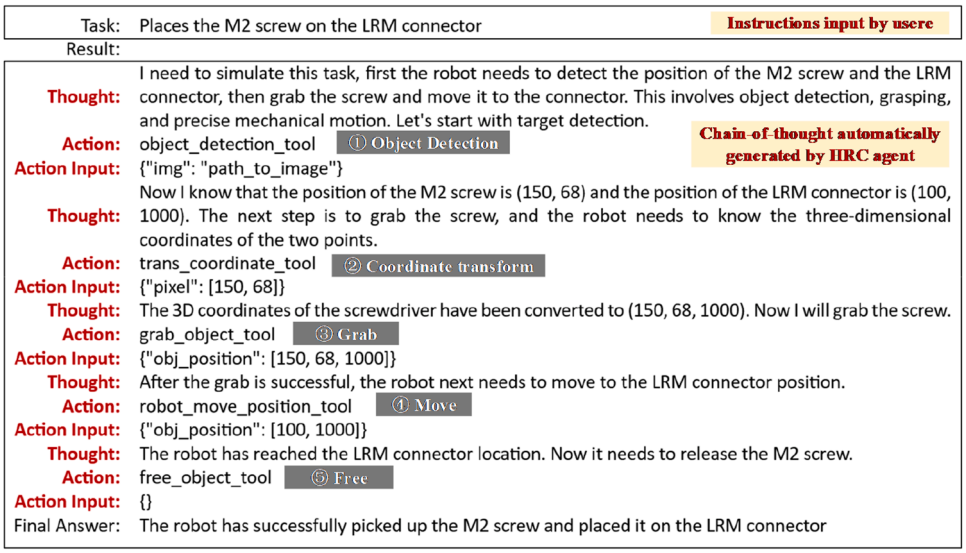


Fig. 12. The working mechanism of agent for task execution.

Table 3
Tasks designed to validate the "task planning success rate" (TPR) and "task execution success rate" (TER).

No. Task	Instruction
2.1	Please hand me a screwdriver
2.2	Grab the wire w1
2.3	Hand me the cable w5
2.4	Put the LRM connector into the chassis
2.5	Put the bracket into the assembly waiting area
2.6	Put the SMA and LRM connectors into the assembly waiting area
2.7	Guide Assembly cable w1 to port SMA1
2.8	Guide Assembly cable w5 to port SMA5
2.9	Guide assembly cable w15 to port SMA13
2.10	Guide assembly cable w13 to port SMA3

Table 4
Results of TPR and TER on designed tasks.

No. Task	Expected action sequence	TPR				TER
		HRC agent (ours)	GLM4	Qwen	GPT4	
2.1	[recognize-move-grab-move-free]	10/10	3/10	5/10	8/10	10/10
2.2	[move-recognize-move-grab-free]	10/10	3/10	1/10	3/10	10/10
2.3	[move-recognize-move-grab-free]	9/10	5/10	1/10	2/10	9/10
2.4	[recognize-move-grab-move-free]	10/10	6/10	5/10	6/10	10/10
2.5	[recognize-move-grab-move-free]	10/10	9/10	8/10	9/10	10/10
2.6	[recognize-move-grab-move-free-move-grab-move-free]	9/10	8/10	8/10	8/10	9/10
2.7	[recognize-move-recognize-move-grab-move-free-move]	10/10	0/10	0/10	10/10	10/10
2.8	[recognize-move-recognize-move-grab-move-free-move]	9/10	0/10	0/10	10/10	9/10
2.9	[recognize-move-recognize-move-grab-move-free-move]	10/10	0/10	0/10	9/10	10/10
2.10	[recognize-move-recognize-move-grab-move-free-move]	10/10	0/10	0/10	10/10	10/10

6.1. Validation of the assembly feature recognizing

To verify target recognition ability, the operator issues an instruction "Pick up the SMA connector". The scene is shown in Fig. 9 and the chain-of-thought is shown in Fig. 10. The SMA connector is placed at a random position in the platform (Fig. 9(a)). The agent first drives the robot to a predefined home position to capture the scene photo (Fig. 9(b)). Then trained YoloV8 model is called by the agent as a tool to detect the SMA connector (Fig. 9(c)) and obtain its pixel coordinates. It can be seen from Fig. 10 that the pixel coordinates of the center position of SMA connector in the image is (801, 419). Then the agent calls the tool of coordinate transformation to converts the pixel coordinates into the robot base coordinate. The output of agent in this step is shown in Fig. 10, which shows the position of SMA in the robot base coordinate system is (−0.25, −0.357, 0.028). Note that the unit of the coordinates is meter. Finally, the robot is driven to move to the target coordinate and then pick up the SMA connector (Fig. 9(e)).

To further test the robustness of HRC agent for assembly feature recognizing and positioning, the following two groups including 17 tasks (tasks 1.1-task 1.17) were designed and each task was repeated 10 times, as reported in Table 2. Group 1 was designed to test the ability of distinguishing different type of objects (screwdriver, bracket, SMA connector, LRM connector, etc.). Group 2 was to identify the different position of same type of object in an assembly array (e.g., cable w1, w2, w3, and the portholes with same diameter but in different positions on the chassis), which is more challenging. Specifically, group 2 is to test the proposed grid matching method for building the assembly array graph and consequently, the ability of HRC agent to understand the assembly array graph.

The validation results of tasks 1.1-task 1.17 is also illustrated in Fig. 11. Take Fig. 11(a) as example, it shows the comparative results of "Task 1.1 (Get a screwdriver)" and "Task 1.2 (Pick up the screwdriver)". This is to prove that the agent is flexible to understand natural language. The operator can give instructions in different spoken ways rather than a few fixed command words, and the agent can understand and respond correctly.

6.2. Validation of task planning ability of HRC agent

The following validation is designed to validate the working mechanism and the task planning functionality of HRC agent. Note that this validation does not involve the robot and only the task planning functionality of the agent is validated in computer. Firstly, the agent receives

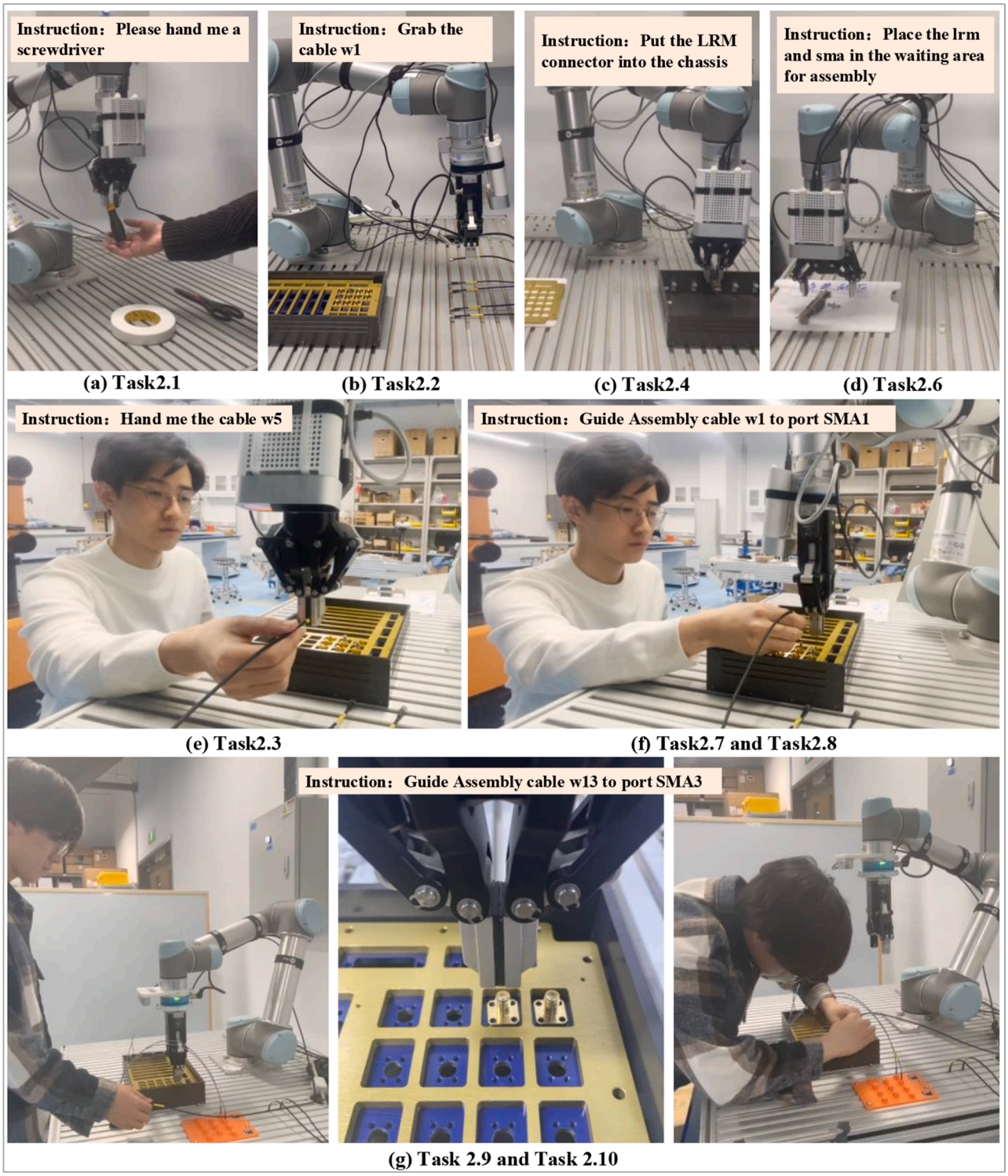


Fig. 13. Verification of Tasks 2.1-2.10.

a text instruction from user: "Place the M2 screws onto the LRM connector". Fig. 12 displays the "Thought", "Action" and "Action input" of the agent during the process of solving this problem. It can be seen that the agent decomposes this task into five sub-tasks. The object detection is firstly carried out to obtain the pixel coordinate of M2 screw and LRM connector. Then the coordinate transformation is implemented to obtain the coordinate of the two parts in the real world. The tool of robotic grabbing is called by the agent, which simulates the robot's action of grabbing screws in the real world. Then the agent guides the robot to move to the target position, and finally commands the robot to release the end gripper and drop the M2 screw onto the LRM connector. It can be seen that the decomposition of the task by the agent is logical and meets the expectations.

In addition, to test the robustness of task planning ability of HRC agent, we designed the following 10 tasks (shown in Table 3) and each task was repeated 10 times. The criterion for task successful is that the planned action sequence output by HRC agent satisfies human expectation. Moreover, HRC agent was compared with three SOTA LLMs (i.e.,

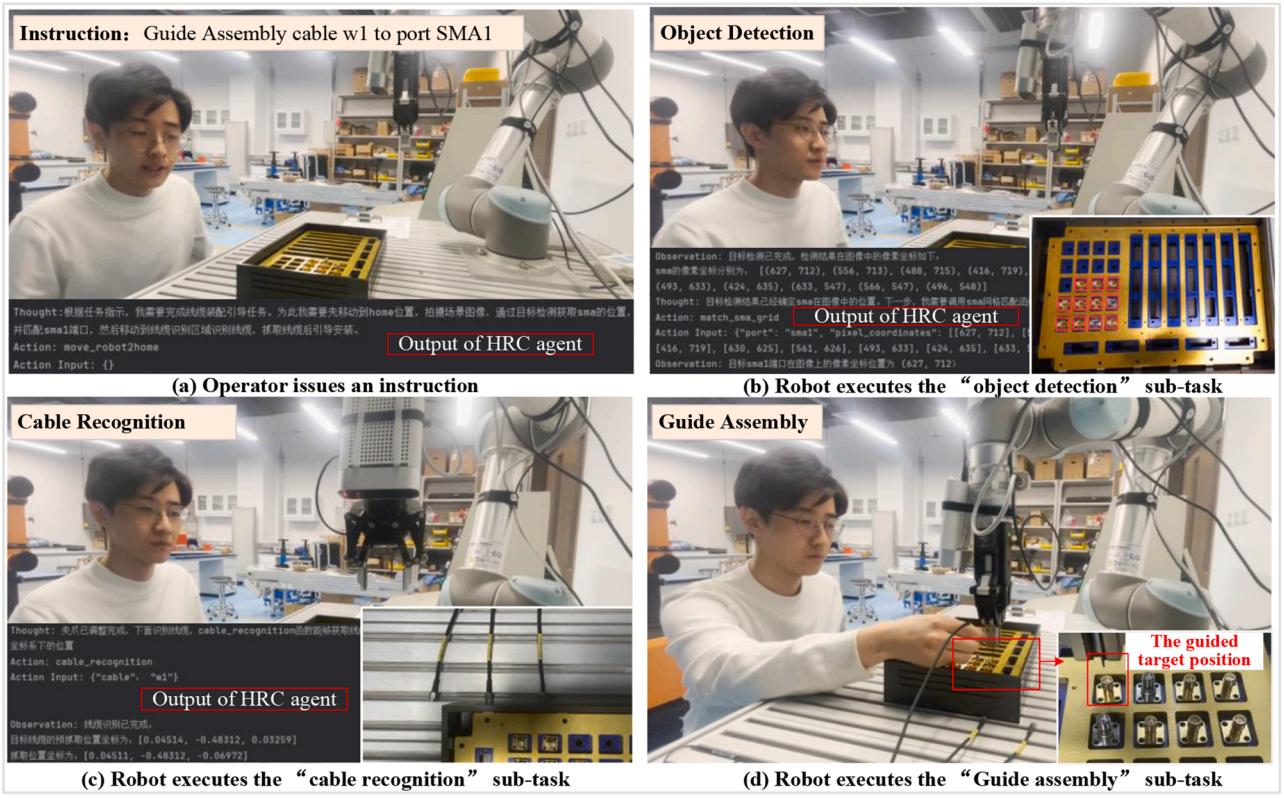


Fig. 14. Whole process of task execution as well as the output of HRC agent in each sub-task.

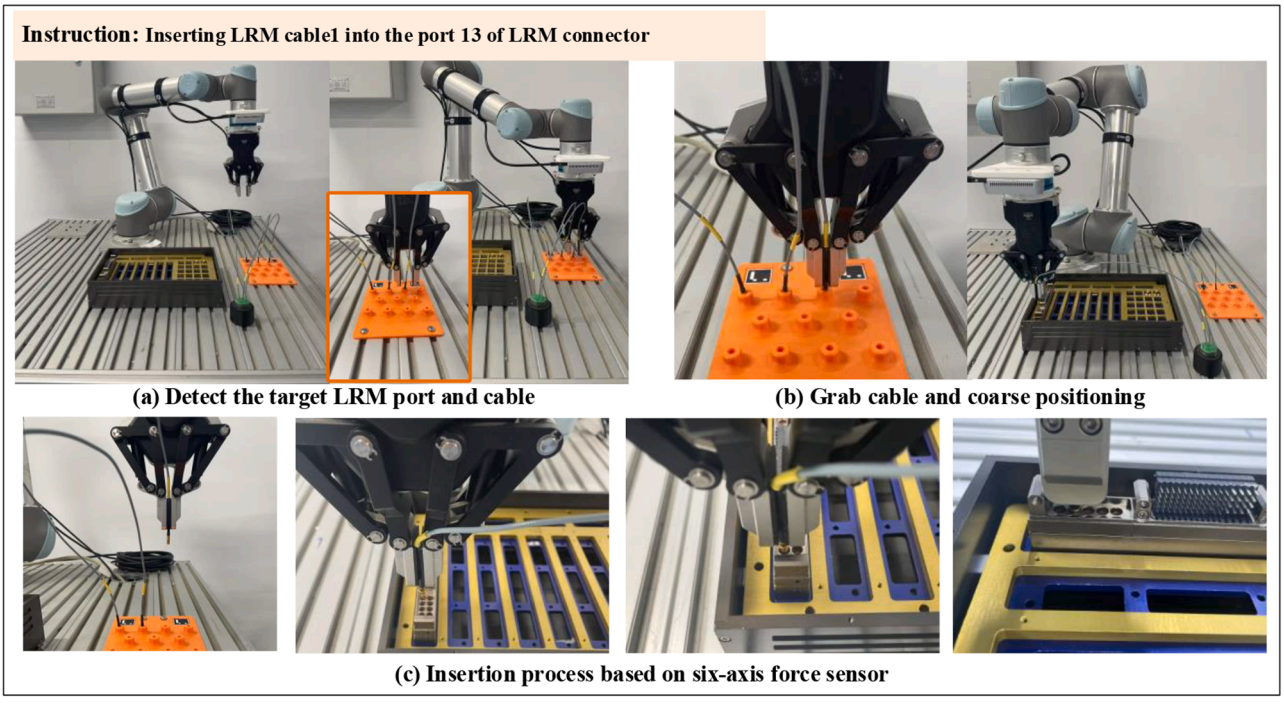


Fig. 15. Generalizability validation of other type of wire assembly.

GLM4, Qwen, GPT4) in terms of task planning ability, as reported in Table 4. The motivation is to evaluate the "chain-of-thought" since the chain-of-thought is the core capability and highlight of the HRC agent. With and without chain-of-thought is equivalent to HRC agent and the general LLMs. For general SOTA LLMs, we proposed a set of prompts to define all available tools and scenario requirements such that the SOTA LLMs can understand which tools can be used to task planning. For each task, the expected action sequence (or human preference sequence) is given, which serves as the ground truth. By comparing the ground truth action sequence with the ones generated by HRC agent and by the three

SOTA LLMs, it can be seen that the proposed HRC agent has a task planning success rate nearly 100 % for the designed eight tasks. In contrast, the three SOAT LLMs can only complete a small part of the planning tasks, i.e., they failed to generate correct and executable action sequences for professional tasks such as wire assembly. The results show that our AI agent is effective and practical for complex tasks planning.

It is noticed that in both Section 6.1 and 6.2 there are a few failure cases. This is due to the LLM-induced hallucinations during task planning process. Since completely eliminating the hallucinations in LLM is very costly, one practical way is to add a rule-based verification module for the task planning result to verify the feasibility of the planning content. Another possible way is to use multi-agent farmwork rather than a single agent since the more knowledge one agent learns, the more likely it is to hallucinate. With multi-agent framework, one can design multiple agents with each of them only taking charge of fine-grained specific responsibilities to avoid hallucinations.

6.3. Validation of task execution ability of HRC agent

On the basis of the task planning verification of Section 6.2, this section continues to use the tasks 2.1–2.10 to validate the execution ability. Note that the difference between Section 6.2 and Section 6.3 are that Section 6.2 does not involve the robot and only the task planning functionality is validated in computer while Section 6.3 needs execution and the robot arm are involved. Each task was repeated 10 times and the criteria for task successful was that all planned actions sequences were correctly executed. The metric TER were calculated and reported in Table 4. The verification scene of each task is shown in Fig. 13.

Based on the planning results of HRC agent, the robot executes the sub-tasks and interacts with the operator. Taking the task of wire assembly guidance as an example, the whole process of task execution as well as the output of HRC agent in each sub-task is shown in Fig. 14. The operator first issues an instruction “Guide assembly cable w1 to the port SMA1”. Then the HRC agent carried out task planning and task decomposition: “According to the instructions, I need to complete the cable assembly guidance task. To do this, I need to first move to the home position, take scene image, obtain the position of SMA through object detection and match the SMA1 port. Then move to the cable recognition area to identify the cable, grab the cable, and guide the installation.”, as shown the “Output of HRC agent” area in Fig. 14(a). Subsequently, the HRC agent drive the robot to implement sub-tasks such as “Objection detection”, “Cable recognition”, and “Guide assembly”, as shown in Fig. 14(b)-(d).

To further test the generalizability of the HRC agent, a new validation of another type of wire assembly is added, i.e., inserting the LRM wire into the hole of LRM connector on the chassis, as shown in Fig. 15. The workflow of HRC agent is similar to Task 2.7-Task 2.10 in Table 3. It should be noted that “Inserting” task is our follow-up work and the detail of the six-axis force sensor-based inserting process is not included in this paper. Inserting the LRM cable into the LRM port (hole) is by the robot is challenging since the cable is very thin with diameter only 2∼3 mm and the LRM hole is with diameter only 4 mm. Imitation learning and reinforcement learning are considered to address the “Inserting” task.

7. Conclusions

This paper proposed HRC agent, which empowers the robot with human’s think mode and execution ability of sensing, interaction, self-reasoning, task planning and task execution. Original LLM without fine-tuned is used as the “brain” of the HRC agent and a series of customized tools are developed as its “hands” to execute tasks. The vision system serves as the “eyes” of HRC agent to capture the environment information. Functional logic and working mechanism of HRC agent is designed such that its “brain”, “eye”, “hands” can work in reasonable coordination.

The chain-of-thought technique that generates a series of intermediate reasoning steps is adopted to improve the ability of LLMs to perform complex reasoning and thus implement complex task. Few-shot learning is used such that HRC agent can quickly learns new specific industry tasks by provided a few examples. The reflection-based contextual memory mechanism enables HRC agent to have long term memory and continuous instruction understanding ability. A series of tools is developed and integrated into HRC agent, by which the capabilities of HRC agent can be easily expanded without much changing of the code framework. The functionality and effectiveness of HRC agent is validated in the aerospace wire harnessing assembly task, whose products has the characteristics of small diameter wires, complicated wire text, dense and tiny assembly holes, varying product batch size and customized production, and thus has high requirements for flexibility. The results show that the HRC agent is able to well understand the natural language instructions and give correct and effective response by chain-of-though, and subsequently, drive the robot to execute tasks by calling tools.

The work will be expanded in two directions. First, we will focus on enabling autonomous understanding of assembly process. Instead of issue voice instruction by operators, assembly process document will be directly input to the LLM. The LLM will then parse and understand the process document and autonomously plan the entire assembly process. Second, we will address the wire insertion challenge. Since the diameters of the wire and the hole are very small (4mm∼7 mm), inserting the wire into the hole entirely by robot is very challenging. Proximal Policy Optimization (PPO) combined with admittance control methods are currently being explored to optimize the inserting process.

CRediT authorship contribution statement

Yiwei Wang: Writing – review & editing, Writing – original draft, Methodology, Conceptualization. **Qi Guo:** Validation, Software, Data curation. **Lianyu Zheng:** Supervision. **Binbin Wang:** Validation, Investigation. **Pai Zheng:** Visualization, Validation. **Zhonghua Qi:** Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This research is funded by the National Natural Science Foundation of China (No. 52375476, No.51805262), Top Young Talents Project of Beihang University (No.YWF-23-L-1209), Beijing Key Laboratory of Digital Design and Manufacturing Project and MIIT Key Laboratory of Intelligent Manufacturing Technology for Aeronautics Advanced Equipment, Ministry of Industry and Information Technology.

Appendix

The developed tools integrated in the HRC agent are listed in Table 5.

Table 5
The developed tools in the tool library.

Category of tool	Name of tool	Function of tool
Tools for processing audio	record_audio	Recording
	my_asr	Speech recognition
	my_tts	Audio synthesis
	play_audio	Audio playback
Tools for processing images captured by depth camera	get_kinect_images	Capture image by depth camera
	object_detection_by_yolo	Object detection
	get_object_depth	Depth value extraction
	cable_recognition	Cable number recognition
	match_sma_grid	Match SMA port
	match_lrm_grid	Match LRM port
	calibration_cam	Hand-eye calibration
Tools for coordinate transformation	trans_pixel2camera	Transformation from pixel coordinate to camera coordinate
	trans_camera2base	Transformation from camera coordinate to robot base coordinate
Tools for robot control	move_robot2home	Robot arm homing
	move_robot2position	Robot arm movement
	move_robot2cable	Robot arm recognizing cable
	grab_object	End gripper grabbing
	grab_cable	End gripper ready to grab cable
	free_object	End gripper release

Data availability

Data will be made available on request.

References

[1] J. Leng, X. Zhu, Z. Huang, et al., Unlocking the power of industrial artificial intelligence towards industry 5.0: insights, pathways, and challenges, J. Manuf. Syst. 73 (2024) 349–363.
[2] F. Sherwani, M.M. Asad, B.S.K.K. Ibrahim, Collaborative robots and industrial revolution 4.0 (IR4.0), in: 2020 International Conference on Emerging Trends in Smart Technologies (ICETST), IEEE, 2020, pp. 1–5.
[3] S. Park, X. Wang, C.C. Menassa, et al., Natural language instructions for intuitive human interaction with robotic assistants in field construction work, Autom. Constr. 161 (2024) 105345.
[4] European Commission, Industry 5.0: towards a sustainable, human-centric and resilient European industry, Policy Brief, Directorate-General for Research and Innovation, 2021.
[5] W. Buerkle, A. Eaton, Al-Yacoub, et al., Towards industrial robots as a service (IRaaS): flexibility, usability, safety and business models, Robot. Comput.-Integr. Manuf. 81 (2023) 102484.
[6] C. Zhang, J. Chen, J. Li, et al., Large language models for human-robot interaction: a review, Biomim. Intell. Robot. (2023) 100131.
[7] D. Andronas, Z. Arkouli, N. Zacharaki, et al., On the perception and handling of deformable objects—a robotic cell for white goods industry, Robot. Comput.-Integr. Manuf. 77 (2022) 102358.
[8] A. Monguzzi, T. Dotti, L. Fattorelli, et al., Optimal model-based path planning for the robotic manipulation of deformable linear objects, Robot. Comput.-Integr. Manuf. 92 (2025) 102891.
[9] S. Liu, L. Wang, X.V. Wang, Multimodal data-driven robot control for human–robot collaborative assembly, J. Manuf. Sci. Eng. 144 (5) (2022) 051012.
[10] X. Zhao, Y. Chen, L. Qian, et al., Human–robot collaboration framework based on impedance control in robotic assembly, Eng. 30 (2023) 83–92.
[11] C. Zhang, G. Zhou, D. Ma, et al., A deep learning-enabled human-cyber-physical fusion method towards human-robot collaborative assembly, Robot. Comput.-Integr. Manuf. 83 (2023) 102571.
[12] V.K.R. Pabolu, D. Shrivastava, M.S. Kulkarni, A digital-twin based worker's work allocation framework for a collaborative assembly system, IFAC-Papers On Line. 55 (10) (2022) 1887–1892.
[13] G. Evangelou, N. Dimitropoulos, G. Michalos, et al., An approach for task and action planning in human–robot collaborative cells using AI, Procedia CIRP. 97 (2021) 476–481.
[14] S. Li, P. Zheng, S. Pang, et al., Self-organizing multiple human–robot collaboration: a temporal subgraph reasoning-based method, J. Manuf. Syst. 68 (2023) 304–312.
[15] T. Bänziger, A. Kunz, K. Wegener, Optimizing human–robot task allocation using a simulation tool based on standardized work descriptions, J. Intell. Manuf. 31 (7) (2020) 1635–1648.
[16] R. Zhang, J. Li, P. Zheng, et al., A fusion-based spiking neural network approach for human–robot collaboration in assembly, Robot. Comput.-Integr. Manuf. 85 (2024) 102634.
[17] R. Zhang, Q. Lv, J. Li, et al., A reinforcement learning method for human-robot collaboration in assembly tasks, Robot. Comput.-Integr. Manuf. 73 (2022) 102227.
[18] P. Zheng, S. Li, L. Xia, et al., A visual reasoning-based approach for mutual-cognitive human-robot collaboration, CIRP Ann 71 (1) (2022) 377–380.
[19] J. Male, U. Martinez-Hernandez, Deep learning based robot cognitive architecture for collaborative assembly tasks, Robot. Comput.-Integr. Manuf. 83 (2023) 102572.
[20] Y. Zhang, K. Ding, J. Hui, et al., Human-object integrated assembly intention recognition for context-aware human-robot collaborative assembly, Adv. Eng. Inform. 54 (2022) 101792.
[21] S. Li, P. Zheng, J. Fan, et al., Toward proactive human-robot collaborative assembly: a multimodal transfer-learning-enabled action prediction approach, IEEE Trans. Ind. Electron. 69 (8) (2021) 8579–8588.
[22] S. Makris, P. Aivaliotis, AI-based vision system for collision detection in HRC applications, Procedia CIRP 106 (2022) 156–161.
[23] C. Li, P. Zheng, S. Li, et al., AR-assisted digital twin-enabled robot collaborative manufacturing system with human-in-the-loop, Robot. Comput.-Integr. Manuf. 76 (2022) 102321.
[24] C.H. Chu, Y.L. Liu, Augmented reality user interface design and experimental evaluation for human-robot collaborative assembly, J. Manuf. Syst. 68 (2023) 313–324.
[25] L. Ouyang, J. Wu, X. Jiang, et al., Training language models to follow instructions with human feedback, Adv. Neural Inf. Process. Syst. 35 (2022) 27730–27744.
[26] A. Kirillov, E. Mintun, N. Ravi, et al., Segment anything, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 4015–4026.
[27] J. Ho, T. Salimans, A. Gritsenko, et al., Video diffusion models, Adv. Neural Inf. Process. Syst. 35 (2022) 8633–8646.
[28] Q. Zheng, X. Xia, X. Zou, et al., Codegeex: a pre-trained model for code generation with multilingual benchmarking on humaneval-x, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 5673–5684.
[29] G. Bian, C. Gao, W. Che, et al., Natural language understanding for Chinese-based service robots, in: 2022 International Conference on Automation, Robotics and Computer Engineering (ICARCE), IEEE, 2022, pp. 1–3.
[30] D. Driess, F. Xia, M.S.M. Sajjadi, et al., Palm-e: an embodied multimodal language model, arXiv preprint arXiv:2303.03378, 2023.
[31] B. Zitkovich, T. Yu, S. Xu, et al., RT-2: vision-language-action models transfer web knowledge to robotic control, in: Conference on Robot Learning, PMLR, 2023, pp. 2165–2183.
[32] W. Huang, C. Wang, R. Zhang, et al., VoxPoser: composable 3D value maps for robotic manipulation with language models, arXiv preprint arXiv:2307.05973, 2023.
[33] W. Huang, F. Xia, T. Xiao, et al., Inner monologue: embodied reasoning through planning with language models, arXiv preprint arXiv:2207.05608, 2022.
[34] I. Singh, V. Blukis, A. Mousavian, et al., ProgPrompt: generating situated robot task plans using large language models, in: 2023 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2023, pp. 11523–11530.
[35] C. Huang, O. Mees, A. Zeng, et al., Visual language maps for robot navigation, in: 2023 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2023, pp. 10608–10615.
[36] D. Shah, B. Osiński, S. Levine, LM-Nav: robotic navigation with large pre-trained models of language, vision, and action, in: Conference on Robot Learning, PMLR, 2023, pp. 492–504.
[37] J. Wu, R. Antonova, A. Kan, et al., TidyBot: personalized robot assistance with large language models, Auton. Robots 47 (8) (2023) 1087–1102.
[38] Y. Obinata, K. Kawaharazuka, N. Kanazawa, et al., Semantic scene difference detection in daily life patrolling by mobile robots using pre-trained large-scale vision-language model, in: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2023, pp. 3228–3233.
[39] C. Gkournelos, C. Konstantinou, S. Makris, An LLM-based approach for enabling seamless human-robot collaboration in assembly, CIRP Ann (2024).

[40] C. Aristeidou, N. Dimitropoulos, G. Michalos, Generative AI and neural networks towards advanced robot cognition, CIRP Ann (2024).
[41] P. Zheng, C. Li, J. Fan, et al., A vision-language-guided and deep reinforcement learning-enabled approach for unstructured human-robot collaborative manufacturing task fulfilment, CIRP Ann (2024).
[42] C. Li, D. Chrysostomou, H. Yang, A speech-enabled virtual assistant for efficient human-robot interaction in industrial environments, J. Syst. Softw. 205 (2023) 111818.
[43] Y. Yin, K. Wan, P. Zheng, An LLM-enabled human demonstration-assisted hybrid robot skill synthesis approach for human-robot collaborative assembly, CIRP Ann.-Manuf. Technol. (2025).
[44] H. Fan, X. Liu, J.Y.H. Fuh, et al., Embodied intelligence in manufacturing: leveraging large language models for autonomous industrial robotics, J. Intell. Manuf. (2024) 1–17.
[45] S. Kannan, V. Venkatesh, B. Min, SMART-LLM: smart multi-agent robot task planning using large language models, in: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2024.
[46] S. Yao, J. Zhao, D. Yu, et al., ReAct: synergizing reasoning and acting in language models, arXiv preprint arXiv:2210.03629, 2022.
[47] J. Liu, Q. Sun, H. Cheng, et al., The state-of-the-art, connotation and developing trends of the products assembly technology, J. Mech. Eng. 54 (11) (2018) 2–28.