Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

RoDA: A Role-Playing Dual-Agent Framework to Drive Nursing Robots in Bimanual Coordination Tasks

Zhendong Zhao

202112801012@stu.hebut.edu.cn

Hebei University of Technology

Yang Li

Hebei University of Technology

Jiexin Xie

Guilin University of Electronic Technology

Tonghui Zhang

Hebei University of Technology

Shijie Guo

Hebei University of Technology

Research Article

Keywords: nursing robot, dual-agent system, LLM role-playing, dialogic co-scheduling, bimanual coordination tasks

Posted Date: September 19th, 2025

DOI: https://doi.org/10.21203/rs.3.rs-7368845/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

RoDA: A Role-Playing Dual-Agent Framework to Drive Nursing Robots in Bimanual Coordination Tasks

Zhendong Zhao[1], Yang Li[1], Jiexin Xie[2], Tonghui Zhang[1], Shijie Guo[1*]

[1*]School of Artificial Intelligence, Hebei University of Technology, Tianjin, 300130, China.
[2*]School of Electronic Engineering and Automation, Guilin University of Electronic Technology, Guangxi, 541004, China.

*Corresponding author(s). E-mail(s): guoshijie@hebut.edu.cn;
Contributing authors: 202112801012@stu.hebut.edu.cn;
liyang@hebut.edu.cn; jxxie20@fudan.edu.cn;
202411201024@stu.hebut.edu.cn;

Abstract

Effective bimanual collaboration represents a significant research objective for Large Language Model (LLM)-driven nursing robots. However, current approaches are constrained by critical limitations: the single-thread LLM task planner (ST-Planner) lacks co-scheduling, while the conventional multi-agent framework (DABICO) suffers from inefficient information interaction between agents, consequently compromising performance in bimanual collaboration tasks. To overcome these limitations, this study introduces Role-playing Dual Agents (RoDA), a novel dual-agent collaboration framework augmented by LLM Role-Playing. This framework implements the nursing robot as an LLM-based dual-agent system wherein each agent assumes the role of either the left or right arm. Through meticulously crafted contextual prompts explicitly defining specific identity attributes and conversational protocols for each limb, these agents facilitate high-quality collaborative dialogue reflective of their designated roles. Evaluation of RoDA was conducted through four MuJoCo simulation scenarios encompassing all four categories of bimanual collaboration, diverse task sequences, and different degrees of workspace overlap. The experimental results demonstrate that the role-playing mechanism enhances dialogue normalization, accuracy, and information richness. This high-quality interaction enables superior

task planning performance, with RoDA surpassing both the ST-Planner and the baseline DABICO framework. Specifically, RoDA achieved a **100%** success rate across all scenarios, exhibiting an average performance improvement of **16.3%** in the Step metric and **70.2%** in the Restep metric relative to DABICO. Furthermore, enhancement of performance is achieved without the requirement for LLM fine-tuning, offering advantages including flexibility, immediacy, and low development cost. Finally, RoDA was demonstrated through practical experiments on a dual-arm nursing robot.

Keywords: nursing robot, dual-agent system, LLM role-playing, dialogic co-scheduling, bimanual coordination tasks

1 Introduction

Dual-arm nursing robots, typically incorporating humanoid torso-mounted limbs and a mobile chassis, demonstrate significant potential for replacing human caregivers in diverse assistance tasks. This capability not only enhances the quality of life of individuals with reduced mobility but also addresses shortages within the caregiving workforce. Execution of tasks such as hygienic cleaning and task assistance necessitates one or more modes of bimanual manipulation by these robotic systems. Current nursing robot systems exhibit limited autonomous capabilities, relying primarily on teleoperation for dual-arm collaboration [1–4]. In this paradigm, the human operator functions as the predominant controller, responsible for task comprehension, decomposition, behavior selection, and execution of motion planning. Large Language Models (LLMs) demonstrate strong capabilities in generalized semantic comprehension and multi-level reasoning for complex semantics. These attributes enable linkage between task objectives and executable actions, thereby empowering robotic systems to understand tasks, perform task decomposition, execute behavior selection, and reduce task-level operator dependence.

However, existing research on LLM-driven robots, exemplified by SayCan [5] and Inner Monologue [6], utilizes a single-thread LLM task planner (ST-Planner) for task planning of individual robotic arms. These ST-Planners lack capabilities for cooperative planning with other agents or knowledge acquisition through information exchange [7]. Consequently, effective deployment of an ST-Planner within dual-arm nursing robots requiring inter-agent (arm) collaboration and information interaction presents significant challenges. Recent studies on LLM-based Multi-Agent Systems (MAS) have demonstrated collaborative task reasoning capabilities among multiple agents through information exchange [8]. This multi-agent methodology exhibits significant advantages over single-agent systems when addressing complex tasks or scenarios [9, 10]. Building upon these findings, our previous work[11] proposed the DABICO framework, an LLM-based dual-agent collaborative framework comprising bimanual robotic task planning via inter-agent information interaction. However, the performance of DABICO system depends critically upon linguistic validity, precision, and informational density within multi-round dialogue content generated by the dual-agent system.

Notably, DABICO lacks comprehension of robotic role characteristics (referring to the distinct functional attributes of each manipulator), rendering high-quality dialogue generation exclusively through task-contextual prompts practically unattainable [12].

Recent investigations into LLM role-playing [13] demonstrate the capability of LLMs to generate high-quality dialogue content depending on pre-defined role characteristics, enhancing task focus [14]. Inspired by these foundations, this work proposes Role-playing-enhanced Dual-Agent framework (RoDA), a dual-agent collaboration framework incorporating LLM role-playing. As shown in Fig. 1. Our core approach assigns manipulator-specific role information within the Dual-Agent system (e.g., Agent1 = Right Arm) to facilitate LLM role-playing in nursing robots. This framework comprises three critical components: 1) Configuration of the dual-arm nursing robot as an LLM-based dual-agent system; 2) Implementation of the dual agents to embody the roles of the left and right arms via LLM role-playing; 3) Establishment of a high-level communication mechanism for information exchange between the agents.

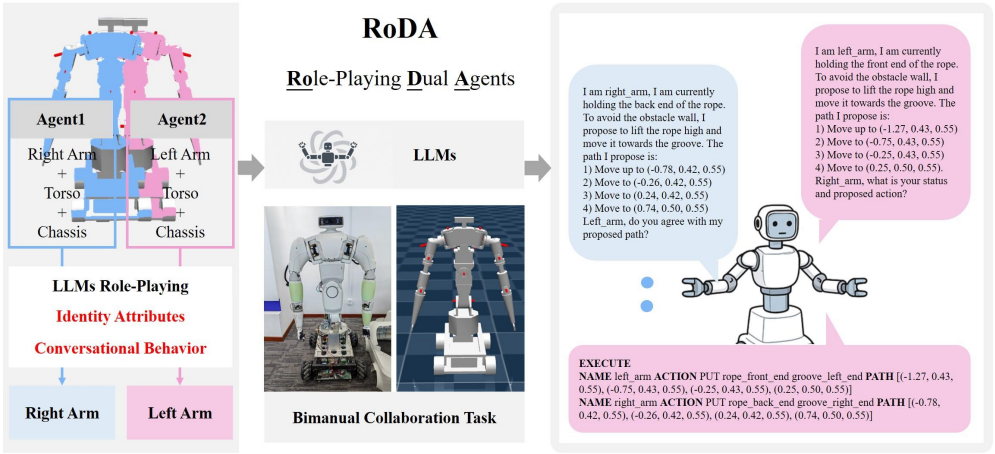


Fig. 1 Enhancing Dual-Agent Collaboration through LLM-based Role-Playing. Bimanual collaboration tasks require effective co-scheduling and high-quality information interaction. RoDA addresses this challenge by conceptualizing the robot as a dual-agent system, where each agent embodies the role of a left or right arm. Through carefully designed contextual prompts, the agents engage in role-playing, generating high-quality dialogue to achieve conversational co-scheduling.

Within this framework, the dialogic co-scheduling between the two agents can be achieved through information interaction. Contextual prompt-based role assignment enables manipulator-specific embodiment (left/right arm), consequently enhancing the quality of the dialogue content. Subsequently, four MuJoCo simulation scenarios were established, featuring diverse task semantics (e.g., sequential compared to parallel execution), varying degrees of workspace overlap, and encompassing all four dual-arm collaboration modes. These scenarios evaluate RoDA performance across multiple bimanual coordination tasks.

In summary, this paper presents three principal contributions:

1. Proposal of RoDA, a novel framework introducing LLM role-playing into dual-arm robotic collaboration. This work pioneers the integration of LLM role-playing into a dual-arm robot collaboration framework, addressing the challenge of low-quality information interaction in dual-agent system and achieving unified co-scheduling and high-quality dialogue.

2. Development and implementation of a dedicated LLMs-role-playing mechanism within RoDA. By leveraging contextual prompts, the dual agents are assigned the roles of the left and right arms, modeling role-specific characteristics and collaborative dynamics to enhance the rationality of task planning.

3. The MuJoCo simulation scenarios encompassing all four dual-arm collaboration modalities, and real-world experiments were conducted on a dual-arm nursing robot. Experimental results demonstrate superior performance of RoDA relative to both the ST-Planner and multi-agent frameworks across bimanual collaboration tasks.

2 Related Work

2.1 LLM-based robots

Currently, the nursing robot system is still constrained by the low level of intelligence and the inability to link the task goal with the operation to be performed. It mainly relies on teleoperation [1–4] to perform bimanual collaboration tasks. The LLM is applied at the task level of the robot system to understand the task, decompose the task and behavior selection, and ultimately address the dependence on the operator at the task level [15]. Such as SayCan [5] and Inner Monologue [6], where LLMs are used for behavior selection and environmental feedback is provided to prompt LLMs for improved task planning. Subsequently, as in Code as Policis [16], ProgPrompt [17], and Demo2Code [18], researchers leverage the code generation capabilities of LLMs to generate high-level instructions into a code format for robotic strategies. The studies such as Text2Motion [19], AutoTAMP [20] and LLM-GROP [21, 22] propose a combination of LLMs with traditional Task And Motion Planning (TAMP). Most previous studies of LLMs-driven robotics adopt a single robotic arm setup for single-thread LLM task planning.

In contrast, the nursing robot is designed to achieve more purposes than the simple superimposition of two robotic arms. Specifically, the nursing robot possesses a symmetrical structure of left and right arms, located on each side of the torso, with a rigorous skill combination, and a high workspace overlap. Both arms in the same system are required to achieve information interaction, coordination and cooperation for completion of the bimanual collaboration tasks. However, the single-thread LLM task planner is incapable to collaborate with other agents in acquiring knowledge. Due to this inherent shortcoming of single-thread LLM task planner, its application in scenarios is restricted, where multiple agents are required to work together and interact with information [7]. Therefore, it is inappropriate to apply a single-thread LLM task planner directly to dual-arm task planning for nursing robot [11, 12].

2.2 Multi-Agents collaboration

Inspired by human group cooperation, researchers have proposed a multi-agent system based on LLM. Such as Camel [14], ChatDev [23], ROCO [24], MetaGPT [25], etc., in which the strategies related to task division of labor and collaboration are explored. Similar to the nature of group cooperation in humans, multi-agent systems leverage the capability of LLMs to generate text for communication for acquiring knowledge through multiple rounds of multi-agent interaction [8]. Multi-agent coordination can be achieved by multiple agents based on their expertise and through information interaction. These capabilities allow multi-agent system to accomplish complex planning and operational tasks by making decisions and planning through the discussion between agents. Compared to single-thread LLM task planners, the multi-agent collaborative approach performs better in solving the tasks that require collaboration between multiple agents [9].

The bimanual coordination task is a typical task where multiple agents are required to work collaboratively. Typical examples include task assistance (handling object with both arms simultaneously) and hygienic cleaning (organizing objects with both arms alternately), etc. Bimanual coordination tasks usually involve one or more forms of dual-arm collaboration where collaborative work and an information interaction between the left and right arms are required. In our previous study [11], dual-arm coordination tasks were addressed by constructing a single entity nursing robot as an LLM-based dual-agent system. Although this framework achieved information interaction and co-scheduling between the left and right arms within the same system, the performance of the LLM-based multi-agent system is directly affected by the quality of the multi-round dialogue content generated by the agents [26]. Meanwhile, the DABICO system struggles to generate high-quality dialogue content using only task-related context prompts, leading to its low performance in handling dual-arm coordination tasks involving handover operations.

2.3 LLMs Role-Playing

The quality of the multi-round information generated by the LLM-based multi-agent system has a direct effect on the reasoning performance [27]. LLMs are trained on large-scale textual datasets to learn about the basic structure and patterns of the language, which enables LLMs role-playing [28]. In the context of LLMs, role-playing refers to the process of LLM generating appropriate dialog content based on predefined role characteristics [10]. LLMs role-playing also allows for the generation of corresponding conversational behaviors, such as following specific conversational rules (i.e., interaction styles) and using specific conversational styles [29]. Research has shown that the role-playing with LLMs enables LLMs to better focus on the task for the generation of normative, accurate, and informative dialog content, thus enhancing the performance in reasoning and decision-making when complex problems are dealt with [14].

LLMs role-playing can be achieved through contextual prompts to improve the performance of the framework. Such as XDAI [30] where the linguistic knowledge and expressive skills of LLMs in relevant domains are activated through the contextual

prompts used to guide LLMs to mimic the behavioral norms of the character. DiscussNav [31] assigns LLM and MLM (Multi-modal Large Model) experts roles and specific tasks promptly to activate their domain knowledge and capabilities. These approaches do not require additional training or fine-tuning, showing such advantages as flexibility, immediacy and low development cost. Inspired by the above studies, our work focuses on the implementation of LLMs role-playing in a dual-agent system through contextual prompts. This aims to improve the quality of multi-round information (dialog content) of the dual-agent system for its better performance.

3 PROPOSED METHOD

This section details the proposed LLM-based dual-agent collaboration framework, explaining how role-playing enhances the quality of multi-round dialogue content between the dual agents. The proposed method is illustrated in Fig. 2.

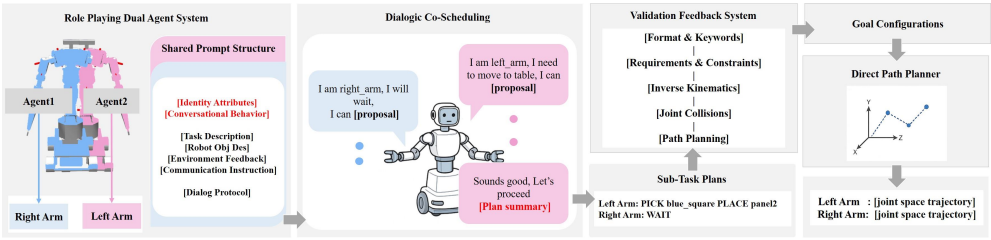


Fig. 2 The RoDA Framework Overview. First, the roles of LLMs are defined for the agents of the left and right arms, respectively. Second, the basis of role-playing in turn drives dialogic co-scheduling between two arm agents and terminate with sub-task plans. Subsequently, the sub-task plans are subjected to a validation feedback system to ensure that their successful execution. Finally, the validated sub-task plans are executed to generate goal configurations for the two arms, with motion trajectories outputted for the nursing robot by the Direct Path Planner.

3.1 Role-Playing Enhanced Dual-Agent Collaboration Framework

It is crucial to achieve both collaborative scheduling and high-quality information interaction between the left and right arms. Based on our previous work[11], the nursing robot is constructed as two collaborative agents (Agent1: right arm, torso, and chassis; Agent2: left arm, torso, and chassis), each equipped with an LLM for dialogue co-scheduling and decision-making. This work introduces LLM Role-Playing (RP) as the key enhancement to improve the quality and effectiveness of multi-round dialogues between the two agents. (Role-Playing is detailed in subsection B.)

To leverage the ability of LLMs to generate text for communication and response, a high-level communication mechanism is established. It enables the dual agents to collaborate on task planning and decision-making through information interaction. A dialog protocol is then developed so that the dual agents can propose "sub-task plans" for the current round within a limited number of discussions. (high-level communication mechanism is detailed in subsection C.) The Context Prompt utilizes a shared

LLM prompt structure, but the specifics are agent-specific and variable depending on the agent role, task requirements, and dual-arm collaboration forms. By introducing role-playing, dialogue coordination between agents can generate "sub-task plans" that are more consistent with their respective roles, capabilities, and constraints. Each agent receives specific information and is required to respond according to its designated role, capabilities, and communication format requirements. At the end of the current round, the last agent to speak summarizes the "sub-task plans" in the required format, which contains information about the sub-tasks of each agents and the 3D waypoints in the task space (in Pack task).

The "sub-task plans" must first be validated through a validation feedback system to ensure their successful execution. The validation is performed in strict order as follows, with the first check passed prior to the next one, until all five validations are passed. 1) Format and Keywords Checking. 2) Requirements and Constraints Checking. 3) Inverse Kinematics solving. 4) Joint Collisions checking. 5) Path Planning Feasibility (in Pack task). If the validation fails, the reason for the failure is recorded and incorporated into the prompt for the next round of dialogue to facilitate replanning. If the task is not completed within the limited number of rounds, it is considered as a failure.

Finally, the validated "sub-task plans" are combined with inverse dynamics to co-generate goal configurations for the left and right arms. For paths defined by waypoints, intermediate goal configurations are determined for each segment. These goal configurations are then passed to the Direct Path planner, which outputs the executed motion trajectories for the left and right arms through linear interpolation between configurations. In addition, the 3D spatial reasoning capability of LLM in the Pack task is explored in this work. Given the start point, goal point, and obstacle location (optional) in the task space, the LLM generates a uniformly distributed 3D waypoint paths by considering the task semantics and constraints comprehensively. This sequence of waypoints defines the intended trajectory, allowing for direct path execution between them, after passing the validation checks.

3.2 Role Playing with LLMs

In this section, it is explained how to implement LLMs role-playing for nursing robot through contextual prompts. Identity Attributes and Conversational Behavior are crucial for LLMs role-playing. Identity attributes provide background information for the nursing robot and determine its physical structure, which influences the content of response from the character. Differently, conversational behaviors exhibit linguistic characteristics and interaction patterns, with effect on conversational style of the character. Therefore, the following LLMs role-play is designed in terms of identity attributes and conversational behaviors by combining the characteristics of nursing robot and bimanual collaboration tasks, as detailed in Fig. 3.

The identity attribute consists of the following key components:

1) Agent Name: To specify the agent corresponding to the left and right arm, and the name of the agent. That is to say, the Agent_name corresponding to the left arm is left_arm, and the Agent_name corresponding to the right arm is right_arm. This

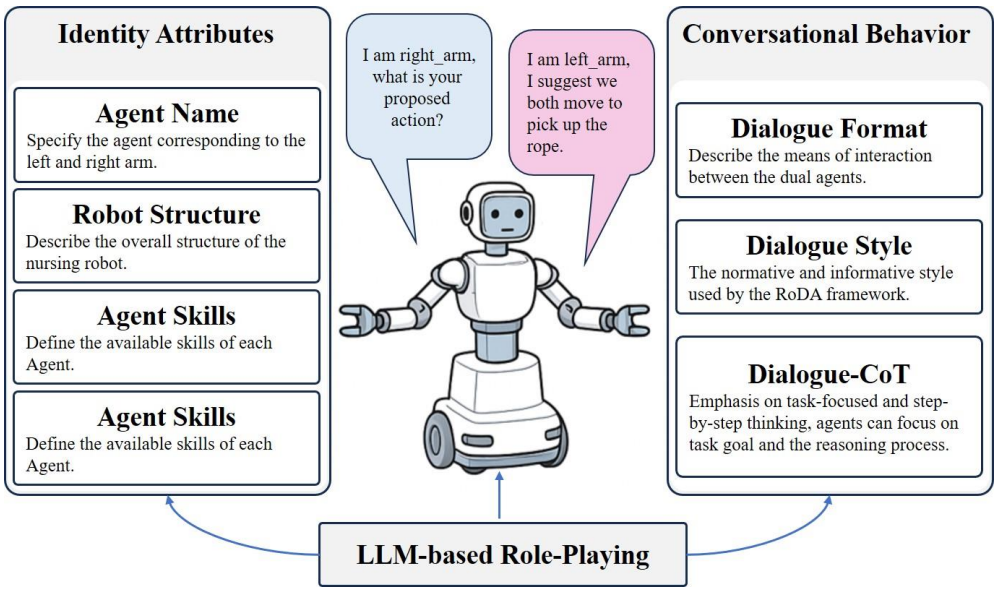


Fig. 3 Details of prompts for LLMs role-playing. RoDA conceptualizes the robot as a dual-agent system, where each agent embodies the role of a left or right arm. Through carefully designed contextual prompts, the agents engage in role-playing, generating high-quality dialogue to achieve conversational co-scheduling. This process enhances task reasoning and improves overall performance in complex bimanual coordination tasks.

ensures clarity and the direct reflection of the role of each Agent, which is conducive to distinguishing between different entities and their respective tasks.

2) Robot Structure: To describe the overall structure of the nursing robot, including the main components, the position of the dual arms, the types of joints, and DoF of each joint. This contextual information provides LLMs with the information about the nursing robot ontology.

3) Agent Skills: To define the available skills of each Agent. Agent skills must be set according to different tasks requirements. During the process of task planning, the LLMs reason about the sub-task plans based on the task requirements, current task progress and available skills.

4) Agent Constraints: Due to the mechanical structure of the nursing robot, both arms need to be taken into consideration to follow the torso and move the chassis to the same side during task planning. Thus, the sub-task planning for the left and right arms becomes more stringent, while the setting of constraints for each agent should comply with different task requirements.

The conversational behavior consists of the following key components:

1) Dialogue Format: To describe the means of interaction between the dual agents. It is ensured that the two agents can have a well-organized and adequate discussion with each other.

2) Dialogue Style: The normative and informative style in which the framework generates dialog content must be maintained. For example, when the Pack task is dealt

with, it is necessary to reason the sub-task plan and 3D waypoints simultaneously. In this case, the dialog content is required to contain the two types of information and to be normative.

3) Dialogue-CoT: To design zero shot Dialogue-CoT for the framework. With emphasis on task-focused and step-by-step thinking, agents can focus on task goal and the reasoning process. The agents are encouraged to continuously monitor and correct errors during the dialog by prompting for error checking and correction.

Overall, RP transforms the two agents from generic communicators into role-embodied collaborators with a shared understanding of the nursing robot's Identity Attributes and Conversational Behavior. As a result, the dialogues evolve beyond simple information exchange into a role-centric reasoning process, where each response is deeply informed by the agent's roles. This foundation of role-playing is crucial for generating the high-quality, actionable sub-task plans required for complex bimanual coordination.

3.3 High-Level Communication Mechanism

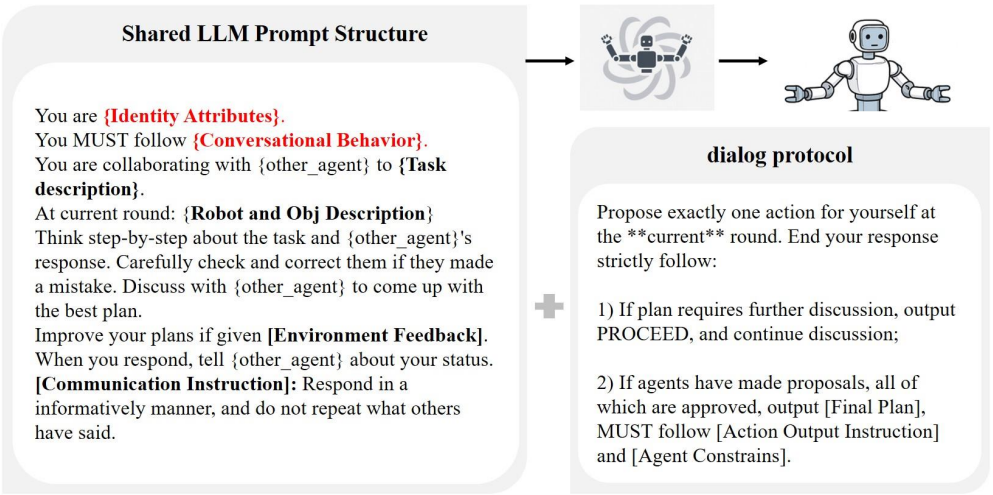


Fig. 4 Shared LLM Prompt Structure. The LLM prompt of each Agent adopts a shared LLM prompt structure, but the content is agent-specific, which varies accordingly to the requirements of the task and the different forms of dual arm collaboration.

This section focuses on the high-level communication mechanism and dialog protocol of the dual-agent collaboration framework, as shown in Fig. 4. The key enhancement lies in the shared LLM prompt structure, which explicitly includes role-playing component to define the agent's identity attributes and conversational behavior. The respective agents of the left and right arms are assigned an LLMs each. Although each agent adopts a shared LLM prompt structure, the content is agent-specific and critically influenced by its role. Before the dual agent interacts with the

environment each time, each agent first calls the LLMs to start a dialog. Each agents receives a specific dialog content and must respond according to its capabilities, communication format, etc. For example, the left arm is incapable to grasp the item on the far right side of the table.

The Shared Prompt consists of the following key components: 1) Task Description: Describe the current task scenario and the final goal of the task. 2) Role Playing: Describe the role information carried by the agent, including identity attributes and conversational behavior. 3) Robot and Obj Description: The status information of each agent and each object at the current round. 4) Environment Feedback: Return the reason for failure in the execution of the previous subtask, e.g. IK failure, collision detected, etc. 5) Communication Instruction: Describe how agents respond to the dialogues of other agents and output the correct format. Subsequently, the dialog protocol is established to ensure that the two agents are free to discuss with each other and formulate a "sub-task plan" for the current round in a limited number of discussions.

3.4 Details on RoDA Algorithm

To formalize this interactive process and evaluate its performance, we implemented the RoDA framework within the Mujoco physics engine. In the simulation experiments, the Mujoco physics engine enables a representation of the environment, including the position, orientation, and contact situation of the robot and objects. By transforming the environment information into structured prompts, RoDA is capable to perceive and understand the current environment information. Subsequently, the RoDA generates sub-task plans through dialogical task collaborative reasoning, which is based on environmental information and shared LLM Prompt. The sub-task plans are then transformed into concrete robot actions and executed after the validation feedback system is passed. The execution results are then used to generate new environment states, and the environment information is incorporated into the next round of LLM prompts until the task is accomplished or the maximum number of task steps is reached.

The pseudo-code is shown in Algorithm 1. where o_t represents the environmental observations, σ_t denotes the joint goal configurations of the robot, and r_t refers to the reward value used to ascertain whether the current task progresses successfully.

4 Experiments

Two types of bimanual collaboration tasks were established in MuJoCo [32], involving four scenarios that cover four different forms of dual-arm collaboration. Then, a simulation model of the nursing robot was constructed using SolidWorks [33]. For the purpose of experimental comparison, three models were constructed, including two without LLMs role-playing: a single-thread LLM task planner (ST-Planner) and a dual-agent collaboration framework (DABICO); and one with LLMs role-playing: RoDA.

The aim of conducting experiment is to answer the following questions when different forms of dual-arm collaboration are dealt with. 1) How is the performance of the RoDA compared to the ST-Planner? 2) How does RoDA's performance improve upon

Algorithm 1 Role-Playing Dual-Agent.

Require: Task horizon T;
Require: RolePlaying_Setup;
Require: Max re-plans K;
Require: Feedback buffer F;
Require: Validation system V;

$t \leftarrow 0$
$o_t \leftarrow$ env.reset()
$Agent_L \leftarrow$ InitAgent(RolePlaying_Setup)
$Agent_R \leftarrow$ InitAgent(RolePlaying_Setup)
while $t < T$ **do**
    F.empty()
    $plan_valid \leftarrow$ false
    while len(F) $< K$ **do**
        $prompt_1 \leftarrow$ GenContext($Agent_L, o_t, F$)
        $prompt_2 \leftarrow$ GenContext($Agent_R, o_t, F$)
        $dialog, plan \leftarrow$ CoSchedule($prompt_1, prompt_2$)
        $plan_valid, feedback \leftarrow$ Validate($plan, V$)
        if $plan_valid$ **then**
            $final_plan \leftarrow$ Parse($plan$)
            break
        end if
        F.append(feedback)
    end while
    if $plan_valid$ **then**
        $\sigma_t \leftarrow$ DirectPathPlanner($o_t, final_plan$)
        $o_{t+1}, r_{t+1} \leftarrow$ env.step(σ_t)
        if $r_{(t+1)} > 0$ **then**
            if IsTaskDone(o_{t+1}, r_{t+1}) **then**
                break
            end if
        end if
        $o_t \leftarrow o_{t+1}$
    end if
    $t \leftarrow t + 1$
end while

the DABICO? 3) Has the quality of dialog content generated by RoDA improved? In what ways?

4.1 Simulation Scenarios

This section elaborates on the setup of simulation scenarios that involve different forms of dual-arm collaboration. To complete bimanual collaboration tasks, nursing robots are usually required to perform one or more forms of dual-arm collaboration.

To simulate bimanual collaboration tasks in the real world, two types of tasks are set up, involving four forms of dual-arm collaboration within a shared workspace: 1) two robotic arms operate the same object at the same time; 2) two robotic arms operate the same object sequentially; 3) two robotic arms operate the different object at the same time; 4) one robotic arm operates an object while the other is in wait for operation. Two types of bimanual collaboration tasks involve a total of four scenarios: Pack-Cube, Pack-Rope, Sort-Two, and Sort-Three, as shown in Fig. 5.

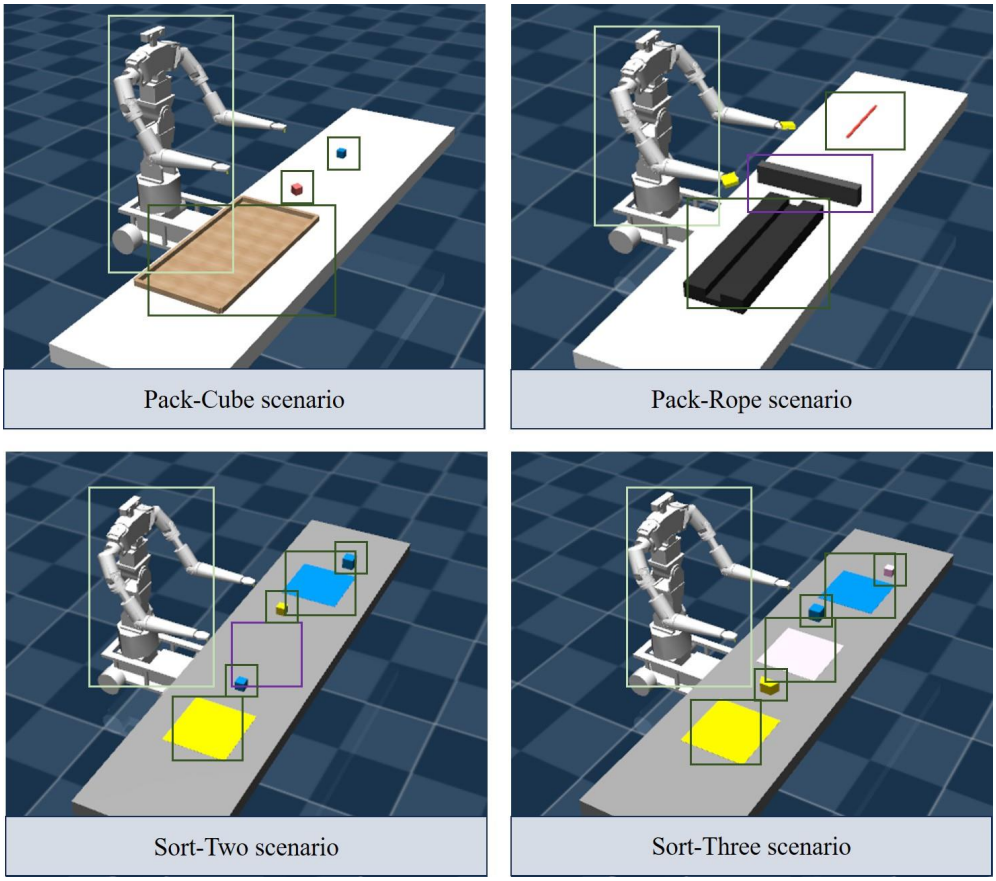


Fig. 5 four scenarios. The Pack task (first row) simulates task assistance (handling object with both arms simultaneously). The Sort task (second row) simulates hygienic cleaning (organizing objects with both arms alternately).

The Pack-Cube scenario simulates the simultaneous manipulation of the same object by both arms; The Pack-Rope scenario simulates the situation where both arms are handling the same object simultaneously; In the Sort-Two scenario, an exchange area is set up in which the left and right arms operate one object alternately, to simulate the sequential operation of the same object by both arms; The Sort-Three scenario

simulates the situation where a single robot arm operates an object while another arm is in wait for operation.

Each type of task contains five key attributes: 1) Task Semantics: Whether the task can be decomposed into multiple sub-tasks that can be completed in parallel or sequentially. 2) Observation Space: The extent to which each agent receives information about the task and the environment. 3) Workspace Overlap: It describes the proximity between the arms when the movement is executed. 4) Collaboration Forms: The forms of dual-arm collaboration involved in the current scenario. The more forms there are, the higher the requirement on task planning capability from the framework. 5) Required Steps: The minimum number of steps required to complete the task.

Table 1 Overview of key properties of scenarios.

Properties	Pack-Cube	Pack-Rope	Sort-Two	Sort-Three
Semantics	Parrallel	Parrallel	Sequential	Sequential
Observation	Shared	shared	Shared	Shared
Workspace	High	High	Low	Low
Collaboration	3	1	2+4	4
Steps	2	2	5	3

As shown in Table 1, Task Semantics: The Pack task requires both arms to operate one or two objects at the same time, the Pack task is a parallel execution of multiple sub-tasks. The Sort task requires both arms to operate the object sequentially, and the Sort task is to execute the sub-tasks sequentially. Observation Space: Since both arms are part of the nursing robot, the task workspace can be observed jointly by the agents of both arms in all four scenarios. Workspace Overlap: The Pack task requires both arms to operate one or two objects at the same time, which causes the high workspace overlap. The Sort task involves only one arm operating an object, with a low workspace overlap. Collaboration Forms: The Pack-Cube scenario and the Pack-Rope scenario involve the third and first forms of dual-arm collaboration, respectively. The Sort task involves the fourth form of dual-arm collaboration. Since the Sort-Two scenario involves handover operation, the second form of dual-arm collaboration is also involved. Required Steps: The Pack task requires both arms to pick_up or put_down one or two objects at the same time, so the minimum steps is 2. The Sort-Two scenario involves a handover operation and requires 5 steps. The Sort-Three scenario requires organizing three squares, with a minimum of 3 steps.

4.2 Experimental setup and evaluation metrics

Two types of bimanual collaboration tasks were established in MuJoCo, involving four scenarios that cover two task semantics (Sequential or Parrallel), two levels of workspace overlap (High or Low), and four forms of dual-arm collaboration. SolidWorks [28] was applied to construct a simulation model of the nursing robot. The model consists of left and right arms, torso, mobile chassis, etc., with a total of 14 DoF. The arms consist of shoulder, elbow and wrist joints, each with 2 DoF. The torso consists of lumbar and hip joints, each with 1 DoF.

OpenAI gpt-4-0613 [34] was adopted for all experiments. The baseline ST-Planner was designed to have complete environmental observation and access to information from both arms, allowing it to generate sub-task plans for both arms simultaneously based on the same planning feedback. To establish a fair comparison, we treat Agent Skills and Agent Constraints as a foundational component for task reasoning. Therefore, we preserve these elements across all methods (including the ST-Planner and DABICO baselines) to maintain functional parity. The key distinction is that RoDA augments these with role-specific contextualization via LLM role-playing, whereas baselines utilize them as static parameters.

Four metrics were created to evaluate the performance of different models: 1) Success: The probability that a task can be completed successfully within ten rounds (i.e., environment steps); 2) Step: The number of rounds spent by the agent in successfully completing an episode. This is a measure of efficiency for the reasoning strategy adopted by the framework; 3) Re_Plan: The average number of re-plans required per turn before an environmental action is executed. This reflects the ability of the agents to understand and use environmental feedback to improve their plans; 4) Re_Step: The number of times a sub-task reasoning is not completed in one turn and the next turn is subsequently initiated. This reflects how efficiently the agent completes the task after reasoning re-starts. Overall, the performance of the method is considered superior when the success rate is higher, the number of environmental steps required is fewer, and the number of replan and re-step is fewer.

We evaluate the quality of the dialogue generated by the agents along three dimensions: 1) Normality: Adherence strictly to the predefined dialogue format and output instructions. This includes clear structural elements (e.g., numbered steps), precise representation of required elements (e.g., coordinates, actions), and the absence of formatting errors or irrelevant content. 2) Accuracy: Factual correctness and logical consistency of the information within the dialogue with respect to the current environment state and robot constraints. This requires the dialogue to contain verifiably correct details (e.g., spatial coordinates, object states, action sequences) and logically coherent step-by-step plans. 3) Information Richness: The level of essential, actionable detail and completeness of the information necessary for making decisions and executing the current sub-task step, with minimal redundancy. This includes providing explicit and precise specifications for actions, objects, locations, and collaborative intent.

4.3 Experimental results

This section evaluates the performance of the three models to answer our research questions. We first compare RoDA directly against the ST-Planner to address question 1, establishing its overall performance advantage. Subsequently, we compare RoDA against DABICO to answer question 2, thereby isolating and validating the specific contribution of the LLM role-playing mechanism.

See table 2, under the Pack-Cube scenario, the task involved the third form of dual-arm collaboration. The models should perform both task planning and path planning. All three models achieved a success rate of 100%, with the best performance achieved by ST-Planner, which was superior to the other two models in all three metrics. RoDA

Table 2 Pack-Cube

	Success	Step	Re_plan	Re_Step
ST-Planner	**100%**	**2.0**	**0.1**	**0.0**
DABICO	**100%**	2.1	1.1	0.1
RoDA	**100%**	**2.0**	0.5	**0.0**

outperformed DABICO in all three metrics of Step, Re_plan, and Re_Step, with an improvement by 5%, 54%, and 100%, respectively. Although Re_plan was slightly higher than ST-Planner, Step and Re_Step were the same as ST-Planner, both of which achieved the best performance.

Table 3 Pack-Rope

	Success	Step	Re_plan	Re_Step
ST-Planner	90%	3.3	4.0	1.3
DABICO	**100%**	3.6	**0.2**	1.6
RoDA	**100%**	**2.2**	0.9	**0.2**

In the Pack-Rope scenario, see table 3, the task involved the first form of dual-arm collaboration, where the left and right arms were required to operate the rope at the same time. It was more difficult to perform task planning and path planning for the left and right arms due to the obstacle wall. Nevertheless, the RoDA and the DABICO still achieved a success rate of 100%, outperforming the ST-Planner. Although the Re_plan metric increased slightly, RoDA was still capable to reason sub-task plan within 1 Re_plan. In Step and Re_Step metrics, RoDA was improved by 38% and 87%, respectively, with the best overall performance produced.

Table 4 Sort-Two

	Success	Step	Re_plan	Re_Step
ST-Planner	70%	7.1	11	2.1
DABICO	70%	6.5	**0.4**	1.5
RoDA	**100%**	**5.1**	1.2	**0.1**

In the Sort-Two scenario, see table 4, the task involved both the 2nd and 3rd forms of dual-arm collaboration, making it much more difficult to perform task planning. Both DABICO and ST-Planner encountered challenges in dealing with this tasks, both of which achieved a success rate of merely 70%. Notably, with the addition of LLMs role-playing, RoDA showed a significant advantage in dealing with this type of task, achieving a success rate of 100%. Meanwhile, an improvement by 21% and 93% was achieved in Step and Re_step, respectively, with a much better performance produced than DABICO and ST-Planner. The Re_plan was elevated to 1.2, which indicates that RoDA performed more re-planning in the discussion.

Table 5 Sort-Three

	Success	Step	Re_plan	Re_Step
ST-Planner	**100%**	**3.0**	0.4	**0.0**
DABICO	**100%**	**3.0**	**0.0**	**0.0**
RoDA	**100%**	**3.0**	0.1	**0.0**

Finally, in the Sort-Three scenario, the task involved a fourth form of dual-arm collaboration. With no requirement on high synergy between the two arms for the task, the difficulty of its task planning is relatively low. The performance of the three models was comparable, with Step and Re_step both achieving the best performance, with an ultimate success rate of 100%. The RoDA, even integrated with LLMs role-playing, failed to further improve its performance for task planning.

Based on the above experimental results, questions 1 and 2 can be answered: When the first two forms of dual-arm collaboration were dealt with, where seamless dual-arm coordination was required, RoDA outperformed both ST-Planner and DABICO. Notably, RoDA achieved a success rate of 100% in Pack-Rope and Sort-Two scenarios, suggesting a clear advantage. When the third and the fourth forms of dual-arm collaboration were dealt with, the high performance of RoDA, DABICO and ST-Planner was comparable to each other.

4.4 Analysis of dialog content

After established RoDA's superior task-planning performance, we now analyze the underlying mechanism by addressing our third research question: the quality of the generated dialogue. Since the ST-Planner does not have an interactive dialogue component, this analysis focuses on the comparison between RoDA and DABICO. This section focuses on analyzing the quality of the dialog content generated by RoDA when the Pack tasks and the Sort-Two scenario are dealt with.

When the Pack tasks are dealt with, the dual agent framework requires careful planning at both the task level and the motion level. Therefore, the dialog content carries two types of information, i.e., sub-task plans and 3D waypoints. At this point, the dialog content is required to be normative, accurate and informative while following the communication format. To ascertain whether LLMs role-playing improves the quality of dialogue content in the above aspects, the dialog content generated by DABICO and RoDA is analyzed.

As shown in Fig. 7, the dialog content of the RoDA allows for normative representation through precise coordinates and clear instructions when locations and sub-task plans are described. A detailed list of six steps, each with precise coordinate locations, illustrates the accuracy of the dialog content. The positional information and movement paths are detailed in the dialogue content, which indicates the richness of information. In contrast, see Fig. 6, the dialog content of the DABICO is relatively vague in describing the path of movement, such as "move towards the groove", without any specific steps provided, which leads to insufficient precision. Moreover, path description is incomplete and not informative.

I am currently holding the front end of the rope. I propose to move the rope over the obstacle wall and place it in the groove on the left end. I will move my gripper to a height of 0.55, then move towards the groove, and finally lower the rope into the groove. Here is my proposed path:

PATH: [(-1.27, 0.43, 0.55), (-0.5, 0.43, 0.55), (0.25, 0.5, 0.55), (0.5, 0.5, 0.36)]

Bob, what do you think about this plan?

I agree with your plan, Alice. I am currently holding the back end of the rope. I propose to move the rope over the obstacle wall and place it in the groove on the right end. I will move my gripper to a height of 0.55, then move towards the groove, and finally lower the rope into the groove. Here is my proposed path:

PATH: [(-0.78, 0.42, 0.55), (-0.25, 0.42, 0.55), (0.5, 0.5, 0.55), (1.00, 0.5, 0.36)]

Let's discuss further if necessary, or we can proceed with this plan.

I agree with your plan, Bob. Your path seems to be collision-free and maintains the distance between our grippers. Let's proceed with this plan.

EXECUTE
NAME Alice ACTION PUT rope_front_end groove_left_end PATH [(-1.27, 0.43, 0.55), (-0.5, 0.43, 0.55), (0.25, 0.5, 0.55), (0.5, 0.5, 0.36)]
NAME Bob ACTION PUT rope_back_end groove_right_end PATH [(-0.78, 0.42, 0.55), (-0.25, 0.42, 0.55), (0.5, 0.5, 0.55), (1.00, 0.5, 0.36)]

Fig. 6 (a) dialog content generated by the DABICO.

I am currently holding the front end of the rope. My gripper is at (-1.27, 0.43, 0.27). The groove's left end is at (0.50, 0.50, 0.36). The obstacle wall's front top is at (0.00, 0.20, 0.50).

To avoid the obstacle wall, I propose to lift the rope high and move it towards the groove. The path I propose is:
1) Move up to (-1.27, 0.43, 0.55)
2) Move to (-0.75, 0.43, 0.55)
3) Move to (-0.25, 0.43, 0.55)
4) Move to (0.25, 0.50, 0.55)
5) Move to (0.50, 0.50, 0.55)
6) Move down to (0.50, 0.50, 0.36)

This path should avoid the obstacle wall and place the rope in the groove.

Right_arm, what is your status and proposed action?

I am currently holding the back end of the rope. My gripper is at (-0.78, 0.42, 0.20). The groove's right end is at (1.00, 0.50, 0.36). The obstacle wall's back top is at (0.00, 0.80, 0.50).

To avoid the obstacle wall, I propose to lift the rope high and move it towards the groove. The path I propose is:
1) Move up to (-0.78, 0.42, 0.55)
2) Move to (-0.26, 0.42, 0.55)
3) Move to (0.24, 0.42, 0.55)
4) Move to (0.74, 0.50, 0.55)
5) Move to (1.00, 0.50, 0.55)
6) Move down to (1.00, 0.50, 0.36)

This path should avoid the obstacle wall and place the rope in the groove.

Left_arm, do you agree with my proposed path? If so, we can proceed to execute our actions.

I agree with your proposed path, right_arm. It seems to be well planned and should avoid the obstacle wall while placing the rope in the groove.

Let's proceed to execute our actions.

EXECUTE
NAME left_arm ACTION PUT rope_front_end groove_left_end PATH [(-1.27, 0.43, 0.55), (-0.75, 0.43, 0.55), (-0.25, 0.43, 0.55), (0.25, 0.50, 0.55), (0.50, 0.50, 0.55), (0.50, 0.50, 0.36)]
NAME right_arm ACTION PUT rope_back_end groove_right_end PATH [(-0.78, 0.42, 0.55), (-0.26, 0.42, 0.55), (0.24, 0.42, 0.55), (0.74, 0.50, 0.55), (1.00, 0.50, 0.55), (1.00, 0.50, 0.36)]

Fig. 7 (b) dialog content generated by the RoDA.

An analysis was conducted on the number of times the quality of the dialogue can be improved in the Pack tasks. The result is shown in Table 6. In ten rounds of task reasoning under the Pack-Cube and the Pack-Rope scenarios, the RoDA framework

improved the dialog quality with a probability of 80% and 70%, respectively. Ultimately, a success rate of 100% was achieved in both scenarios. According to the results, the improvement in dialogue quality of the RoDA framework is reasonble, showing a certain level of robustness.

Table 6 Percentage of Dialogue Quality Improvement

RoDA	Trial Results (Y/N for 10 runs)					percentage	Success
Pack-Cube	1:Y	2:Y	3:N	4:Y	5:N	80%	100%
	6:Y	7:Y	8:Y	9:Y	10:Y		
Pack-Rope	1:Y	2:N	3:Y	4:Y	5:Y	70%	100%
	6:N	7:N	8:Y	9:Y	10:Y		

Note: Y = yes (successful), N = no (unsuccessful)

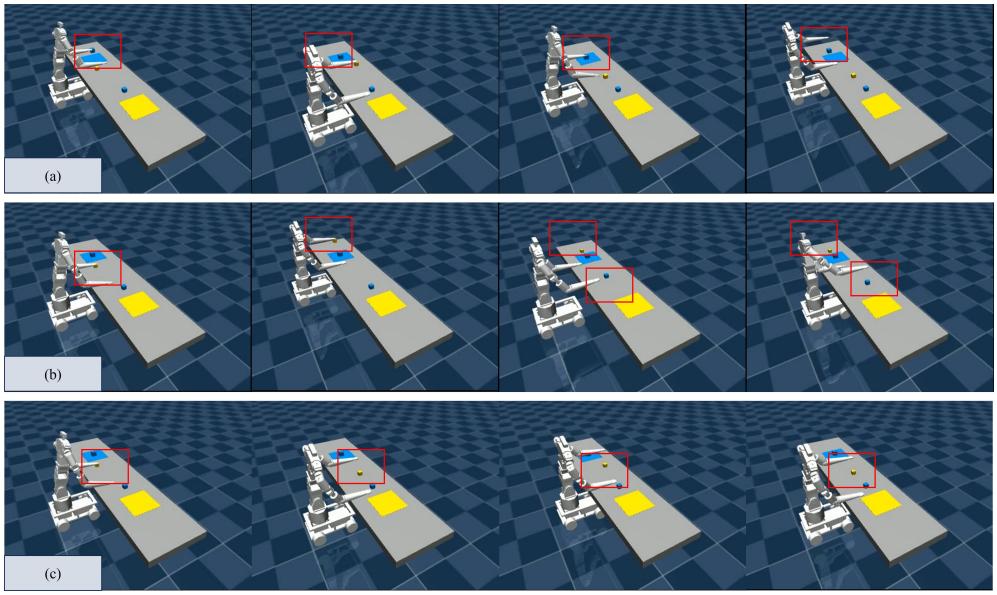


Fig. 8 Failure case. failure case of DABICO framework in Sort-Two scenario. The DABICO tends to get stuck in specific subtask loops and ultimately fails the task.

When Sort-Two scenario is dealt with, the dual agent framework is required to reason the handover operation of two arms, with a sharp rise in the difficulty of task planning. Therefore, the two arm agents are required to be accurate in the dialogue content to reason sub-task plans, thus improving the performance of task planning. It was analyzed how RoDA can avoid inaccurate reasoning information in the dialog content under the DABICO framework.

The DABICO framework failed to complete the task three times, as shown in Fig. 8 (a), (b) and (c). 1) After placing the left blue cube in the blue area (at which point the sub-task was completed), the left arm then placed it on the left of the blue area, repeating the action; 2) The left arm failed to place the yellow cube in the exchange area as required. The right arm then repeatedly performed the action of placing the right blue cube in the exchange area and subsequently taking it out; 3) The left arm repeatedly performed the action of placing the left yellow cube in the exchange area and subsequently taking it out. It is suggested that the DABIO struggled to deal with the second form of dual-arm collaboration. In contrast, RoDA is capable to reason the sub-task plan and complete the task with a success rate of 100%. No specific sub-task loops occur in the process.

Based on the above analyses, question 3 can be answered: The dialogue content generated by RoDA shows robustness with normality, accuracy and information richness, with the probability of dialogue quality improvement stabilized at 70% to 80%. Notably, the improvement of dialogue content quality further enhances the performance of the dual agent framework in task planning. In particular, when the second form of dual-arm collaboration is dealt with, where it is significantly difficult to perform task planning, RoDA achieves a success rate of 100%, which is superior to the other two models.

4.5 Real-world experiment

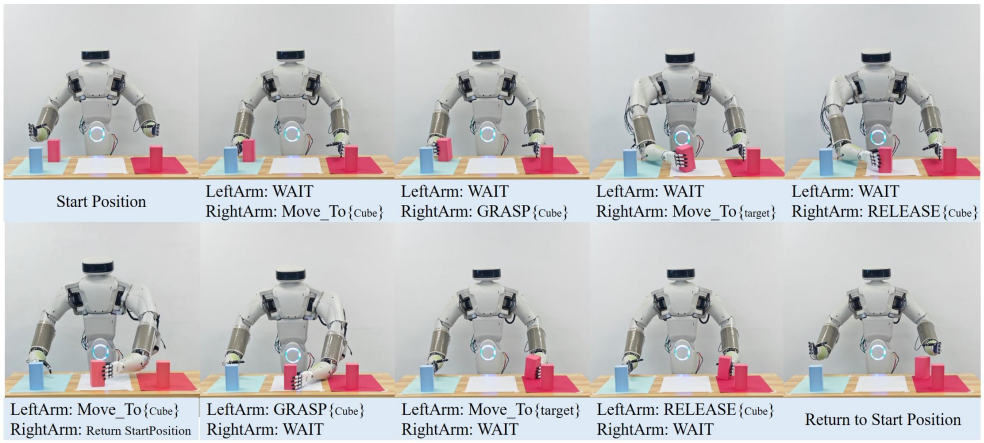


Fig. 9 Cube organization. This task involves the the second and third forms of dual-arm two-armed collaboration where the sequential execution of four eight steps is required (including GRASP, MOVE-TO, RELEASE, and WAIT operations). Under the RoDA framework, co-scheduling and high-quality information interaction can be achieved to generate the sub-task plans that lead to the sequential execution of all four steps.

A real-world experiment replicating the Sort-Two scenario design was conducted, encompassing dual-arm collaboration modes II and III. The nursing robot platform

utilizes an RGB-D camera for environmental perception. Object categories and spatial coordinates are acquired via the implementation of YOLO algorithm. Within the RoDA framework, environmental data undergoes transformation into structured prompts for processing. Subsequent dialog task planning is performed by remotely querying the OpenAI GPT-4-0613 pre-trained model to generate sub-task plans. The sub-task plans are then translated into concrete robot actions through invocation of pre-trained skill primitives. The pre-trained skill primitives, including GRASP and RELEASE, are acquired via teleoperation. During task execution, the adoption of the RoDA framework enables dynamic generation of bimanual sub-task plans, executing all eight sequential steps in orderly fashion as illustrated in Fig. 9. This experiment further confirms the practical efficacy of the RoDA framework.

5 Conclusion

This work introduces RoDA, a novel Role-playing-enhanced Dual-Agent collaboration framework addressing limitations of existing LLM-based systems for dual-arm nursing robots. The core of RoDA enables effective embodiment of the left- and right-arm roles through specialized agent assignments. Integration of a role-playing mechanism via contextual prompts resolves persistent challenges of low-quality dialogue and co-scheduling failures inherent in multi-agent systems. This optimization improves the quality of their interaction, producing dialogue that is normative, accurate, and information-rich. As a result, the collaborative task planning and co-scheduling performance exhibit substantial enhancement. Comprehensive evaluation encompasses four simulation scenarios representing all four forms of bimanual collaboration. Experimental results demonstrate that RoDA achieves a superior overall performance compared to both DABICO and the ST-Planner. Furthermore, real-world experiment on a dual-arm nursing robot confirm practical implementation feasibility. A principal advantage of RoDA lies in achieving these performance gains without expenditure on costly fine-tuning of the underlying LLM. Crucially, exclusive reliance on contextual prompts delivers an adaptable, computationally efficient, and resource-minimal solution.

References

[1] Iwata, H., Sugano, S.: Design of human symbiotic robot twendy-one. In: 2009 IEEE International Conference on Robotics and Automation, pp. 580–586 (2009). IEEE

[2] Guo, S., Shiraoka, T., Inada, S., Mukai, T.: A two-ply polymer-based flexible tactile sensor sheet using electric capacitance. Sensors **14**(2), 2225–2238 (2014)

[3] Ding, J., Lim, Y.J., Solano, M., Shadle, K., Hu, J.: Giving patients a lift - the robotic nursing assistant (rona). In: 2014 IEEE International Conference on Technologies for Practical Robot Applications (TePRA) (2014)

[4] Chung, W.K., Yang, Y., Cui, N., Qian, H., Xu, Y.: Design of a rescue robot with a wearable suit augmenting high payloads rescue missions. In: 2017 2nd

International Conference on Advanced Robotics and Mechatronics (ICARM), pp. 704–711 (2017). IEEE

[5] Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Fu, C., Gopalakrishnan, K., Hausman, K., et al.: Do as i can, not as i say: Grounding language in robotic affordances. arXiv preprint arXiv:2204.01691 (2022)

[6] Huang, W., Xia, F., Xiao, T., Chan, H., Liang, J., Florence, P., Zeng, A., Tompson, J., Mordatch, I., Chebotar, Y., et al.: Inner monologue: Embodied reasoning through planning with language models. arXiv preprint arXiv:2207.05608 (2022)

[7] Talebirad, Y., Nadiri, A.: Multi-agent collaboration: Harnessing the power of intelligent llm agents. arXiv preprint arXiv:2306.03314 (2023)

[8] Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N.V., Wiest, O., Zhang, X.: Large language model based multi-agents: A survey of progress and challenges. arXiv preprint arXiv:2402.01680 (2024)

[9] Blanco-Fernández, D., Leitner, S., Rausch, A.: Interactions between dynamic team composition and coordination: an agent-based modeling approach. Review of Managerial Science **19**(1), 1–37 (2025)

[10] Florez-Lozano, J.M., Caraffini, F., Parra, C., Gongora, M.: A robust decision-making framework based on collaborative agents. IEEE Access **8**, 150974–150988 (2020)

[11] Zhao, Z., Yue, X., Xie, J., Fang, C., Shao, Z., Guo, S.: A dual-agent collaboration framework based on llms for nursing robots to perform bimanual coordination tasks. IEEE Robotics and Automation Letters **10**, 2942–2949 (2025)

[12] Xie, J., Chen, Z., Zhang, R., Wan, X., Li, G.: Large multimodal agents: A survey. arXiv preprint arXiv:2402.15116 (2024)

[13] Shao, Y., Li, L., Dai, J., Qiu, X.: Character-llm: A trainable agent for role-playing. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 13153–13187 (2023)

[14] Li, G., Hammoud, H., Itani, H., Khizbullin, D., Ghanem, B.: Camel: Communicative agents for" mind" exploration of large language model society. Advances in Neural Information Processing Systems **36**, 51991–52008 (2023)

[15] Firoozi, R., Tucker, J., Tian, S., Majumdar, A., Sun, J., Liu, W., Zhu, Y., Song, S., Kapoor, A., Hausman, K., *et al.*: Foundation models in robotics: Applications, challenges, and the future. The International Journal of Robotics Research **44**(5), 701–739 (2025)

[16] Liang, J., Huang, W., Xia, F., Xu, P., Hausman, K., Ichter, B., Florence, P., Zeng, A.: Code as policies: Language model programs for embodied control. In: 2023 IEEE International Conference on Robotics and Automation (ICRA), pp. 9493–9500 (2023). IEEE

[17] Singh, I., Blukis, V., Mousavian, A., Goyal, A., Xu, D., Tremblay, J., Fox, D., Thomason, J., Garg, A.: Progprompt: Generating situated robot task plans using large language models. In: 2023 IEEE International Conference on Robotics and Automation (ICRA), pp. 11523–11530 (2023). IEEE

[18] Wang, H., Gonzalez-Pumariega, G., Sharma, Y., Choudhury, S.: Demo2code: From summarizing demonstrations to synthesizing code via extended chain-of-thought. In: Proceedings of the 37th International Conference on Neural Information Processing Systems, pp. 14848–14956 (2023)

[19] Lin, K., Agia, C., Migimatsu, T., Pavone, M., Bohg, J.: Text2motion: From natural language instructions to feasible plans. Autonomous Robots **47**(8), 1345–1365 (2023)

[20] Chen, Y., Arkin, J., Dawson, C., Zhang, Y., Roy, N., Fan, C.: Autotamp: Autoregressive task and motion planning with llms as translators and checkers. arXiv preprint arXiv:2306.06531 (2023)

[21] Ding, Y., Zhang, X., Paxton, C., Zhang, S.: Task and motion planning with large language models for object rearrangement. In: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2086–2092 (2023). IEEE

[22] Zhang, X., Zhu, Y., Ding, Y., Zhu, Y., Stone, P., Zhang, S.: Visually grounded task and motion planning for mobile manipulation. In: 2022 International Conference on Robotics and Automation (ICRA), pp. 1925–1931 (2022). IEEE

[23] Qian, C., Cong, X., Yang, C., Chen, W., Su, Y., Xu, J., Liu, Z., Sun, M.: Communicative agents for software development. arXiv preprint arXiv:2307.07924 **6** (2023)

[24] Mandi, Z., Jain, S., Song, S.: Roco: Dialectic multi-robot collaboration with large language models. arXiv preprint arXiv:2307.04738 (2023)

[25] Hong, S., Zheng, X., Chen, J., Cheng, Y., Wang, J., Zhang, C., Wang, Z., Yau, S.K.S., Lin, Z., Zhou, L., et al.: Metagpt: Meta programming for multi-agent collaborative framework. arXiv preprint arXiv:2308.00352 (2023)

[26] Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., et al.: Autogen: Enabling next-gen llm applications via multi-agent conversation. arXiv preprint arXiv:2308.08155 (2023)

[27] Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., *et al.*: A survey on large language model based autonomous agents. Frontiers of Computer Science **18**(6), 186345 (2024)

[28] Shanahan, M., McDonell, K., Reynolds, L.: Role play with large language models. Nature **623**(7987), 493–498 (2023)

[29] Zhou, J., Chen, Z., Wan, D., Wen, B., Song, Y., Yu, J., Huang, Y., Peng, L., Yang, J., Xiao, X., et al.: Characterglm: Customizing chinese conversational ai characters with large language models. arXiv preprint arXiv:2311.16832 (2023)

[30] Yu, J., Zhang, X., Xu, Y., Lei, X., Guan, X., Zhang, J., Hou, L., Li, J., Tang, J.: Xdai: A tuning-free framework for exploiting pre-trained language models in knowledge grounded dialogue generation. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 4422–4432 (2022)

[31] Long, Y., Li, X., Cai, W., Dong, H.: Discuss before moving: Visual language navigation via multi-expert discussions. In: 2024 IEEE International Conference on Robotics and Automation (ICRA), pp. 17380–17387 (2024). IEEE

[32] Todorov, E., Erez, T., Tassa, Y.: Mujoco: A physics engine for model-based control. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5026–5033 (2012). IEEE

[33] Matsson, J.: An Introduction to SOLIDWORKS Flow Simulation 2023. SDC publications, ??? (2023)

[34] OpenAI, A.J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. 2023. arXiv preprint arXiv:2303.08774

Declarations

- Funding. This work was supported by National Natural Science Foundation of China under Grant 62303154.
- Competing interests. The authors have no relevant financial or non-financial interests to disclose.
- Author contribution. All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Zhendong Zhao, Yang Li, Jiexin Xie, Tonghui Zhang and Shijie Guo. The first draft of the manuscript was written by Zhendong Zhao and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.
- Ethics approval. This is an observational study. The XYZ Research Ethics Committee has confirmed that no ethical approval is required.

- consent to participate. Informed consent was obtained from all individual participants included in the study.
- Consent for publication. The authors confirm.
- Data availability.
- Materials availability.
- Code availability. The authors confirm.