TOWARDS HUMAN-CENTRIC SMART MANUFACTURING: A SYSTEMATIC VISION-LANGUAGE MODEL-BASED METHODOLOGY FOR HUMAN-ROBOT INTERACTIVE, TASK PLANNING, NAVIGATION, AND MANIPULATION

WANG TIAN

PhD

The Hong Kong Polytechnic University

2026

The Hong Kong Polytechnic University

Department of Industrial and Systems Engineering

Towards Human-Centric Smart Manufacturing: A Systematic Vision-Language Model-based Methodology for Human-Robot Interactive, Task planning, Navigation, and Manipulation

Wang Tian

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

August 2025

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

WANG Tian

Abstract

Rooted in the principles of Industry 5.0, Human-centric Smart Manufacturing (HSM) prioritizes human welfare within manufacturing processes, leveraging Artificial Intelligence (AI) to simultaneously improve operational efficiency and employee well-being. Under this paradigm, Human–Robot Interaction (HRI) technologies are central because they unify human problem-solving and decision-making strengths with the precision and efficiency of robotic systems, resulting in adaptive, intelligent, and superior manufacturing processes.

Nevertheless, existing HRI works commonly face several challenges: 1) Data scarcity for specific tasks - limited availability of data for particular tasks can hinder the development of efficient models or systems; 2) Requirement for re-training for different scenarios - many existing approaches necessitate re-training or fine-tuning models for different scenarios, which can be time-consuming and impractical; 3) Predominantly pre-defined manners - most interaction approaches are predetermined, offering limited adaptability and failing to support open-ended or natural exchanges between humans and robots..

Recently, Vision-Language Models (VLMs) and Large Language Models (LLMs) have attracted significant attention in the field of HRI, and they hold strong potential as effective solutions to address the current limitations faced in HRI. Firstly, as pretrained on vast repositories of world knowledge, VLMs and LLMs possess the capacity to generalize across diverse scenarios and tasks, thereby addressing the limitations of previous models that suffered

from insufficient datasets and required frequent retraining, thus rendering HRI more adaptable and effective across multiple domains. Secondly, VLMs strengthen robotic intelligence by fusing visual and linguistic inputs with unified reasoning, thereby mirroring the multimodal perception characteristic of humans. Thirdly, VLMs and LLMs enhance user experience by enabling flexible interaction and communication, thereby making HRI more natural, efficient, and intuitive.

In this case, this research aims to develop a VLM-based HRI methodology that makes it possible for robots to engage with humans naturally and systematically follow their commands to collaborate on accomplishing complex tasks—such as supplementing assembly materials, repairing malfunctioning machines, and detecting potential safety issues—toward future human-centric smart manufacturing. To realize this goal, three fundamental robotic tasks warrant exploration: robot system task planning, mobile robot navigation, and robot arm manipulation. Accordingly, this research proposes a systematic vision-language model-based methodology for human-robot interactive task planning, navigation, and manipulation in manufacturing.

Chapter 2 provides a survey of vision–language techniques applied to HRI within human-centered smart manufacturing, and concludes by outlining unresolved challenges in task planning, navigation, and manipulation.

In Chapter 3, the background and research challenge of vision and language-based task planning are introduced, and a coarse-to-fine vision-grounded Chain-of-Thought (CoT) reasoning approach is proposed to address the unforeseen challenge faced by smart manufacturing systems. By integrating multi-granularity visual parsing with a coarse-to-fine CoT framework, our method jointly leverages global scene context and local visual perspectives to enable interpretable, zero-shot reasoning, while generating clear and precise robot-executable task plans. The evaluation results indicate that our framework attains remarkable success and generalization in challenging

as well as novel scenarios, emphasizing its reliability and applicability to real-world smart manufacturing environments.

In Chapter 4, the background and research gaps of vision and language navigation is introduced. Then proposes a human-guided mobile robot navigation method for unstructured manufacturing environment. Finally, we use some experiments to evaluate these methods. Learning-based VLN methods heavily rely on data and are constrained by the availability of a fixed number of scenes for training. This limitation prevents them from effectively addressing the challenges of realistic application scenarios, where objects may exist outside the pre-defined domains and language instructions can be more diverse. Additionally, although the current method shows promising performance in simulated environments, it faces challenges when applied in real-world scenarios due to sensor noise. This noise can lead to the loss of geometric details in the reconstructed maps, resulting in inaccurate segmentation of the navigation target. To tackle these issues, we integrate classical 3D reconstruction techniques with zero-shot VLM, aiming to construct an open-vocabulary 3D semantic mapping framework. Furthermore, an LLM is utilized to interpret natural language commands and produce control code for robots, thereby facilitating navigation supported by vision and language.

In Chapter 5, the background and research challenge of vision and language-based robot manipulation is fistly introduced, and then a foundational model-based framework is proposed for 3D Hierarchical Affordance Reasoning, concentrating on high-precision tool operations in factory environments. Within this framework, two components are proposed: 3D Joint Tool-Part Affordance Grounding enables open-vocabulary 3D affordance detection, jointly grounding tool and part interactions; task-aware Trajectory Generation leverages an LLM alongside motion affordance priors to generate task-specific motion trajectories, enabling robots to perform complex tool-use operations with high precision and adaptability. Extensive experiments show that the proposed hierarchical affordance reasoning framework demonstrates sub-

stantial improvements over current leading approaches. Experiments carried out on actual manufacturing systems confirm the effectiveness and stability of the proposed approach, highlighting its suitability for real-world industrial settings that require strong dexterity, logical reasoning, and generalization capabilities.

It is hoped that this work can create a multi-robot intelligent system capable of achieving human-centric natural visual-language interaction. Under human guidance, this system will be able to accomplish various tasks in complex factory environments in future human-robot symbiotic manufacturing paradigms.

Keywords: Smart Manufacturing, Human-Robot Interaction, Vision-Language Model, Large Language Model, Robot Task Planning, Robot Navigation, Robot Manipulation

Acknowledgement

Firstly, I would like to express my deepest and most heartfelt gratitude to my supervisor, Prof. Pai Zheng. From the very beginning of my PhD journey, Prof. Zheng has shown unwavering patience and dedication in guiding and inspiring me to identify a meaningful research direction. Despite his busy schedule, he always made time for our discussions, offering insightful feedback even when my questions were trivial or nascent. What I cherish most is how he respected my research interests, consistently encouraging me to pursue the topics that genuinely ignited my curiosity. Beyond his academic mentorship, Prof. Zheng has generously provided invaluable advice regarding my future career. He has been open and sincere in sharing his personal experiences and recommendations, which have profoundly influenced my professional path. I consider myself immensely fortunate and honored to have had such an exceptional mentor throughout this academic journey. His guidance has not only shaped this thesis but also left a lasting imprint on my growth as a researcher and individual.

Furthermore, I would like to extend my sincere thanks to all the members of the RAIDS research group and fellow students from the ISE Department whom I have been fortunate to know. They have generously shared their experiences and knowledge, offering unwavering academic support whenever I needed guidance. Whether addressing major challenges or minor questions, they were always patient and willing to engage in thoughtful discussion. Their companionship has not only enriched my academic journey but also made my time in Hong Kong vibrant and unforgettable. I am deeply grateful

for the camaraderie and kindness they have shown me throughout these years.

Thirdly, I would like to express my heartfelt gratitude to Dr. Josie Hughes, my mentor during my exchange at EPFL in Switzerland, as well as all the wonderful members of the CREATE LAB. Despite the differences in our research backgrounds, which could have made my time there quite challenging, Josie and everyone in the lab were incredibly patient and supportive. The six months I spent with them were filled with invaluable learning. Everything in Switzerland was wonderful—from brainstorming ideas together and conducting experiments, to hiking in the mountains and sharing drinks. I was especially moved by the heartfelt farewell wishes and gifts they prepared before I left. I feel truly blessed to have shared such a paradise-like time with them.

Lastly, I wish to express my deepest and most heartfelt thanks to my family—my grandparents, my parents, and my fiancé. Throughout my PhD journey, they have given me boundless patience, understanding, and unwavering support. Their love and care have consistently lifted my spirits, reminding me not to dwell on setbacks, but to cherish the many beautiful things this world and humanity have to offer. I feel truly blessed beyond measure to have each of them in my life.

Contents

List of Figures

List of Tables

Nomenclature

2D	Two-Dimensional
3D	Three-Dimensional
AGVs	Automated Guided Vehicles
AI	Artificial Intelligence
CoT	Chain-of-Thought
HRC	Human–Robot Collaboration
HRI	Human–Robot Interaction
HSM	Human-centric Smart Manufacturing
ICP	Iterative Closest Points
IEEE	Institute of Electrical and Electronics Engineers
IMU	Inertial Measurement Units
IoU	Intersection over Union
IO	Input-Output
JTPA	Joint Tool-Part Affordance

LLMs	Large Language Models
PCA	Principal Component Analysis
PDDL	Planning Domain Definition Language
RL	Reinforcement Learning
ROI	Region-Of-Interest
SAM	Segment Anything Model
TaTG	Task-aware Trajectory Generation
TSDF	Truncated Signed Distance Function
VLA	Vision-Language-Action
VLMs	Vision-Language Models
VLN	Vision and Language Navigation

1 Introduction

1.1 Background

Anchored in the ethos of Industry 5.0, Human-centric Smart Manufacturing (HSM) propels human welfare to the forefront of the production sphere, bolstering system performance and human well-being through the adoption of Artificial Intelligence (AI) [1]. This emphasis on human necessities and interests shapes the course of Human-Robot Interaction (HRI). Crucial to HSM, HRI harnesses collaborative human and robotic capabilities to engineer intelligent manufacturing systems prioritizing human requirements [2]. The focus of HRI lies in the design of cooperative robots working alongside humans, thereby boosting productivity and mitigating human strain [3]. Technological strides in AI, robotics, soft materials, and bioelectronics have inextricably entwined HRI within the industrial fabric.

In various industries within the manufacturing sector, there is a growing trend towards mass personalization. This requires smart manufacturing capabilities to efficiently produce personalized products with dynamic batch sizes [4]. In order to achieve this, interaction among multiple intelligent agents is essential. However, recent manufacturing systems lack the flexibility and adaptability to address the changing production demands and unforeseen internal system contingencies, and HRC is a key approach to solving this problem, with human-robot interaction being an indispensable component [1].

Nevertheless, current research on HRI [5, 6, 7, 8] still encounters several critical challenge: 1) Task-specific data scarcity – for many specialized man-

ufacturing scenarios, datasets are either limited or unavailable, making it difficult to train robust and efficient models; 2) Scenario dependence and re-training – existing approaches often require extensive re-training or fine-tuning when transferred to new environments, which reduces scalability and practical applicability; 3) Rigid, pre-defined interaction patterns – most interaction frameworks rely on pre-scripted commands or structured templates, which constrain flexibility and prevent seamless, natural, or free-form communication between humans and robots.

Recently, Vision-Language Models (VLMs) and Large Language Models(LLMs) have gained substantial interest as an emerging research area. LLMs, by harnessing their extensive world knowledge and powerful reasoning capabilities, can generalize effectively to open-domain settings, while VLMs further leverage the pretrained LLM backbone to address multimodal tasks [9]. Advancements in foundation model-based vision and language interaction have the following advantages for HRI: 1) VLMs demonstrate strong versatility as comprehensive task-solvers. VLMs are capable of handling a diverse range of multimodal tasks due to their inherent ability to process and generalize across both visual and linguistic information; 2) The integration of vision and language within VLMs mirrors human perceptual processes more closely. Human cognition naturally relies on multiple senses, which operate in a complementary and cooperative manner. By incorporating multimodal information, VLMs enhance their intelligence through an approximation of this multisensory perception. 3) VLMs also provide an intuitive and accessible interface by enabling flexible modes of interaction and communication. Through multimodal input support, users can engage with intelligent systems in more adaptive and personalized ways, thereby enriching the overall HRI efficiency.

In this case, my research focus on Vision-Language Model-based Natural Human-Robot Interaction for Human-centric smart manufacturing. More specifically, I would like to explore a multi-robot system centered on human

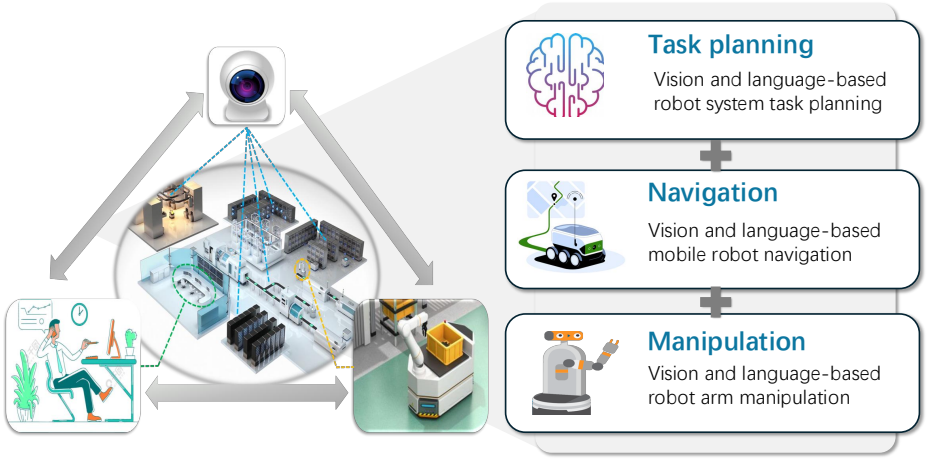


Fig. 1.1: Research scope of the PhD thesis.

language command, integrated with environment understanding, to achieve natural and efficient manufacturing task fulfillment in unstructured senarios. Currently, for most robot system, task completion will follow this pipeline: first performing global task planning, then navigating to the target area, and finally perform manipulation to complete the task. Therefore, three core robotic capabilities are prioritized for investigation: robot system task planning, mobile robot navigation, and robot arm manipulation. Building upon these foundations, this research introduces a systematic VLM-based framework for human–robot interactive task planning, navigation, and manipulation within manufacturing contexts.

1.2 Research scope

To achieve a holistic VLM-based human-robot interaction system in human-centric smart manufacturing, three core tasks should be considered: task planning, navigation, and manipulation, as demonstrated in Fig. 1.1.

Section one addresses task planning for robots using vision and language modalities input, which involves global perception of manufacturing space and executable plans generation for multiple related robots, especially for unforeseen scenarios. Large vision models are utilized for zero-shot visual perception based on global and local views. The captured visual information is passed to VLMs for further reasoning to figure out the region of interest and related robots, and finally generate robotic executable plans.

The second part involves visual and language-based human-guided navigation for mobile robots. Based on the task plans generated in the previous step, mobile robots need to navigate to different locations to perform corresponding operations. Specifically, LLM is used to decompose the task plans, expressed in natural language, into navigation sub-goals and translate them into robot code. Simultaneously, VLM is used to build a 3D semantic map to accomplish the navigation task.

The third part is about the vision and language-based manipulation for robot arms, which involves executing complex manufacturing tasks, such as using some tools for assembly for disassembly, based on the requirements specified in the task plans. LLM and VLM are able to utilize their understanding of the environment to further plan the trajectory of the robot's end-effector. This enables the robot to perform precise and intricate movements necessary to accomplish the specific manipulation tasks.

1.3 Research objectives

In order to address the industrial challenges mentioned above and achieve a holistic VLM-based human-robot interactive system in human-centric smart manufacturing, this project primarily focuses on exploring three main areas: task planning, navigation, and manipulation, as shown in 1.2. The specific research objectives are as follows.

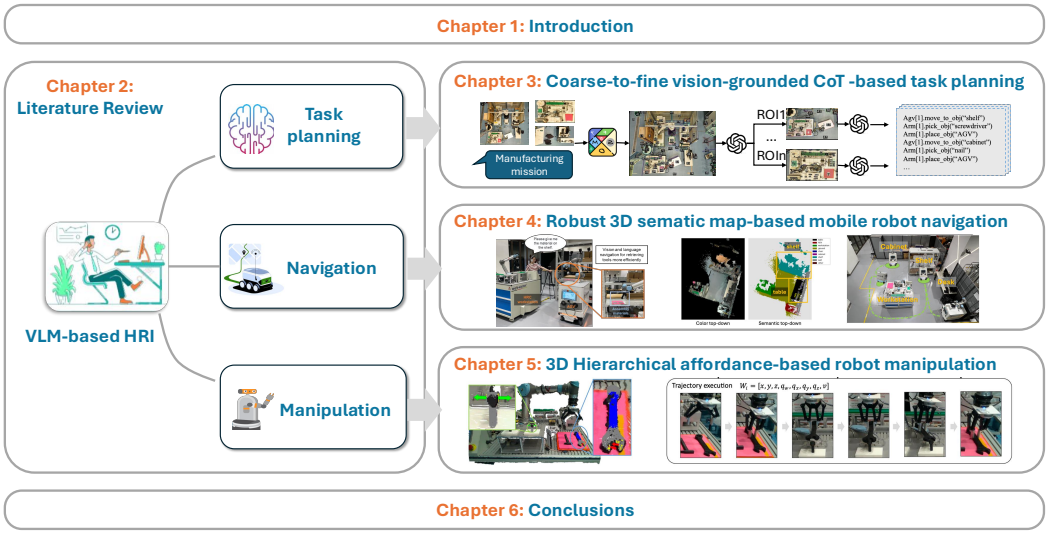


Fig. 1.2: Research objectives of the PhD thesis.

Objective 1: to propose a coarse-to-fine vision-grounded CoT-based task planning method for unforeseen manufacturing scenarios.

Existing approaches to global planning frequently suffer from a high-level, abstract understanding of tasks, whereas detailed planning is typically restricted to predefined, robot-specific action primitives. This critical gap impedes the creation of flexible, executable, and interactive plans for diverse multi-agent systems and dynamic environments. Moreover, despite employing multi-modal (visual and textual) reasoning, plans generated by LLMs and VLMs often contain hallucinations and lack precision, resulting in instructions that are too vague for direct robotic execution. To fill the gap, we aim to propose a coarse-to-fine vision-grounded CoT-based task planning method for unforeseen manufacturing scenarios, integrating joint global-local reasoning and precise visual information, thereby enabling holistic multi-robot task planning.

Objective 2: to introduce a vision and language-based human guided robotic navigation method in human-centric smart manufacturing with extra consideration of sensor noise problems and free-form interaction manner.

Learning-based VLN methods heavily rely on data and are constrained by the availability of a fixed number of scenes for training. This limitation prevents them from effectively addressing the challenges of realistic application scenarios, where objects may exist outside the pre-defined domains and language instructions can be more diverse. Additionally, although the current method shows promising performance in simulated environments, it faces challenges when applied in real-world scenarios due to sensor noise. This noise can lead to the loss of geometric details in the reconstructed maps, resulting in inaccurate segmentation of the navigation target. In order to address these challenges, we propose the fusion of traditional 3D reconstruction methods with zero-shot VLM to establish an open-vocabulary 3D semantic map. Additionally, we leverage LLM to comprehend natural language instructions and generate code for controlling robots, enabling the realization of vision and language-based navigation.

Objective 3: to develop a vision and language-based 3D hierarchical affordance reasoning approach for industrial robot tool-use manipulation.

Current methodologies primarily concentrate on pick-and-place operations for basic industrial functions such as assembly, material handling, and sorting; however, they exhibit limited proficiency in executing complex tool-use operations, including nail hammering, bolt fastening, or screw driving. To address the problems discussed, we aim to propose a foundational model-based 3D hierarchical affordance reasoning approach for robotic tool-use manipulation in industrial tasks. Specifically, a novel VLM with multi-dimensional feature fusion will be introduced to jointly infer graspable and functional affordances based on task semantics and the 3D geometric relationships between tools and parts. In addition, an LLM will be prompt to translate tool-part affordances and linguistic instructions into a series of physical constraints, thus generating post-contact end-effector trajectories as movement affordances.

1.4 Research structure

The rest content of this report is organized as follows:

Chapter 2 provides an overview of vision–language techniques applied to HRI within human-centric smart manufacturing. The chapter concludes by outlining the remaining challenges in task planning, navigation, and manipulation.

Chapter 3 firstly introduces the background and research challenge of vision and language-based task planning. Then proposes a reasoning method to realize holistic multi-robot task planning for unforeseen scenarios. Extensive experimental evaluations are conducted to demonstrate the effectiveness and novelty of the proposed approach.

Chapter 4 firstly introduces the background and research gaps of vision and language navigation. Then propose a framework including 3D reconstruction and semantic segmentation to realize robust 3D semantic segmentation in unstructured manufacturing environment. Finally, the experiments are conducted to evaluate the innovation and efficiency of the proposed method.

Chapter 5 firstly introduces the background and research challenge of vision and language-based robot manipulation, and then provides a 3D hierarchical affordance reasoning approach to realize complex and precise robot manipulation tasks. Finally several experiments are provided and discussed to validate the proposed method.

Chapter 6 summarizes the contributions, limitations, and future works.

2 Literature Review

2.1 Multimodal human-robot interaction

2.1.1 Human-robot interaction in human-centric smart manufacturing

HRI is a research domain probing human-robot linkages that aims to innovate robots that can synergize with humans to enhance efficiency and mitigate human exertion [3]. With its applications sprawling across sectors such as healthcare, tourism, hospitality, surgery, construction, agriculture, and smart manufacturing, the HRI focus elegantly pivots based on the unique demands of each field.

HRI plays a cardinal role in service and social robots within the healthcare, tourism, and hospitality sectors. Healthcare positions social robots as surrogate family members and caregivers, extending support to patients with dementia, autism, and other mental or physical conditions [10]. Concurrently, the tourism and hospitality spheres leverage service robots to enhance customer experiences [11]. Conversely, fields like surgery, construction, agriculture, and smart manufacturing primarily utilize HRI for Human-Robot Collaboration (HRC) and collaborative robots. The surgery field observes surgical robots' advent, offering heightened precision and spatial dexterity, facilitating safer, quicker, and less invasive interventions with improved surgical outcomes [12]. In construction, collaborative robots execute tasks like excavation and assembly, merging worker expertise with autonomous efficiency [13]. Agriculture utilizes HRI for targeted crop recognition, aiming

to enhance agricultural adaptability and efficiency [14]. In smart manufacturing, the aim is to integrate AI in a human-centric way to improve capabilities and create personalized products, a step towards Industry 5.0 [2]. This paper focuses on industrial HRI, specifically multimodal technologies.

HSM is one of the most important characteristics of Industry 5.0, it prioritizes the human at the center of the production system, improves worker health and safety conditions through a synergistic combination of humans and machines, supports individual needs and factory requirements for flexibility, agility, and robustness, focuses on human needs and interests, shifts from a technology-centric to a human-centric and societal-centric approach, and leverages AI thinking methodologies to support decision making to improve system performance and human well-being [1]. In this context, collaborative intelligence is essential for HSM [2]. It necessitates leveraging the capabilities of humans and robots to establish smart manufacturing systems that prioritize human needs, and HRI is the pivotal factor in achieving this outcome. Human beings possess qualities such as leadership, creativity, versatile problem-solving abilities, and decision-making abilities. Meanwhile, robots excel in speed, endurance, quantitative accuracy, scalability, and a vast knowledge base. In a cohesive human-robot system, humans serve as trainers and commanders of robots, playing the role of creators and decision-makers. Meanwhile, robots enhance human cognitive skills and physical capabilities, resulting in a mutual increase in capabilities through HRI that enables them to accomplish complex tasks together [2, 15].

2.1.2 Human-robot interaction with cognitive intelligence

Industrial HRI usually occurs when human operators and robots share a workstation and communicate [16]. The goal of HRI research is to boost production efficiency and ease the workload by creating cooperative robots

that fuse their abilities with human skills [3]. According to IEEE, robots are autonomous machines capable of real-world perception, computation, and action [17]. Commonly employed robots for interaction and collaboration include robotic arms, Automated Guided Vehicles (AGVs), and chatbots. With progress in areas like AI, soft materials, and bioelectronics, the popularity of HRI is surging, drawing numerous researchers from related fields.

In cognitive science, human intelligence is broadly classified into *perception*, *cognition*, and *action* categories [18], as shown in Figure 2.1. *Perception* refers to acquiring information about the surroundings, people, and objects, followed by interpreting the collected sensory signals. This process encompasses gathering information via sensory modalities like vision, audio, and touch, and processing it for recognition and prediction. *Cognition* involves reasoning, making recommendations based on perceived information and assigned goals, and communicating with others. This includes procedures such as planning, problem-solving, and learning. *Action* encompasses the execution of tasks and interactions with people and surroundings, relying on both sensory input and cognitive processes. This includes object manipulation, collaborative tasks, and physical actions. Equipping robots with maximal degrees of these intelligence types will edge us closer towards effectuating more natural HRI.

2.1.3 Multimodal communication and control for natural HRI

The three levels of natural HRI involving cognitive robot intelligence all mandate multimodal information analysis, as shown in Figure 2.1. At the robot perception level, multimodal sensing and interpretation complement each other, providing more accuracy and robustness than single-modality perception. At the cognitive level, multimodal information processing enhances robots' comprehension of their environment and human behavior, making

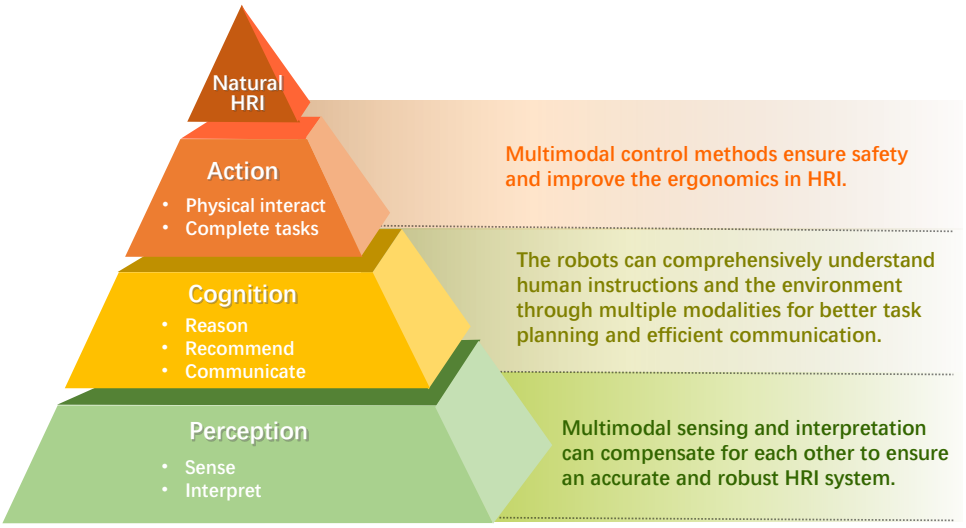


Fig. 2.1: The importance of multimodal technologies for robot cognition intelligence.

task planning and execution more proficient. It also improves human-robot communication efficiency and efficacy. Lastly, multimodal robot control at the action level ensures human interaction safety and enhances ergonomics. Overall, for natural HRI and advancing HSM, multimodal technology is crucial. HRI inclusive of multimodal communication and control methods represents a crucial foundation for future HRI development, supporting a safe, natural, efficient, and intelligent human-robot collaborative system.

2.2 VLM-based HRI

The most crucial modalities for HRI tasks are vision and language, encompassing planning, navigation, and manipulation. Vision equips the robot with person, object, and environment recognition capabilities, while humans employ text or verbal communication (language and auditory modes) to interact with the robotic system. The emergence of LLMs and VLMs, with their rich knowledge bases and strong contextual reasoning, has brought HRI into a transformative stage. First, their extensive knowledge base supports a comprehensive and nuanced understanding of a wide range of subjects, enabling informed decision-making and adaptive contextual reasoning. Sec-

ond, the multimodal perception capabilities of VLMs—integrating visual and linguistic information—significantly enhance robotic intelligence, allowing for more advanced environmental interpretation and interaction. Finally, their sophisticated natural language processing abilities facilitate intuitive human-machine communication by accurately interpreting and generating human-like responses, thereby improving both the efficiency and user experience of HRI applications. Leveraging the advanced capabilities of LLMs and VLMs, our goal is to establish a future human-machine symbiotic smart manufacturing environment, in which three fundamental robotics tasks are included, task planning, navigation, and manipulation. This section discusses the most representative work of LLMs and VLMs for HRI in the three core tasks.

2.2.1 VLM-based HRI for task planning

The remarkable evolution of LLMs and VLMs has enabled the extraction of extensive commonsense knowledge from heterogeneous web-based data sources. This accumulated knowledge base presents substantial opportunities for embodied intelligent agents, facilitating the generation of executable task plans that align with human instructions expressed through natural language interfaces. Typically, in HRI scenarios within manufacturing environments, task planning plays multiple critical roles, including task understanding and decomposition, task allocation, and robot action generation. Most of the existing work in this domain is based on the GPT series models. Some approaches utilize prompt-based direct Input-Output (IO) methods, while others adopt COT reasoning and its variants to enable deeper contextual understanding. Additionally, certain works integrate structured databases and planning algorithms to compensate for the inherent inaccuracies of LLMs. A summary of recent and significant contributions in this field is presented in Table 2.1.

Table 2.1: VLM-based HRI for task planning

Method	Features	Performance	Applications	Year	Ref
LLM-based manufacturing execution system	Behaviour tree, task allocation	100% (format accuracy), 86.67% (executability), 93.33%(goal accuracy)	HRC assembly	2024	[19]
LLM-enabled human demonstration-assisted hybrid robot skill synthesis approach	Direct-IO, task decomposition and allocation	95.92%(average success rate)	HRC assembly	2024	[20]
HRC-adapted LLM	Error correction control module, HRC dataset fine-tuned LLM, action generation	78.57% (human intention accuracy), 75.0% (tool name accuracy), 82.14% (tool feature accuracy)	HRC assembly	2024	[21]
Vision AI-based HRC assembly approach	Direct-IO, task decomposition and action generation	\	HRC assembly	2024	[22]
Vision-language-guided robotic action planning approach	Direct-IO, task decomposition and action generation	93.3% (average success rate)	HRC assembly	2024	[23]
Adaptive intelligent generation method	Dynamic knowledge evolution mechanism, action generation	91.54%(supported collaboration model), 89.31%(simultaneous collaboation model), 90.78%(sequential collaboration model)	Assembly solution generation	2025	[24]
CogniVera	Dual-agent architecture with internal dialogue mechanism, task understanding and decomposition	96.67% (task completion rate)	HRC assembly	2025	[25]
PlanCollabNL	LLM reasoning combine with PDDL, task allocation	84% (subgoals to favour completion rate), 86% (subgoal to disfavour completion rate)	Adaptive plan generation in HRC	2025	[26]
VLM-based human-guided mobile robot navigation approach	Direct-IO, task decomposition and action generation	92.5% (average success rate)	HRC assembly	2025	[27]

The most common approach to leveraging LLMs or VLMs for task planning in HRI tasks involves providing a prompt that contains primitive information, clearly specifying the desired output content and format. The LLM or VLM then directly generates the corresponding output based on this input. Fan et al. [23] leveraged LLMs to understand human input instructions and perform task decomposition. Based on a set of pre-defined interfaces representing robot action primitives, the LLM was able to generate Python code corresponding to a sequence of actions. This code could then be executed by the robot to complete manipulation tasks. The approach achieved a task success rate of 93.3% based on GPT-4. Similarly, Liu et al. [22] and Wang et al. [27] investigated human-guided navigation tasks for mobile robots in assembly environments. They leveraged LLMs to interpret natural language instructions and perform task decomposition. Utilizing a predefined set of navigation action primitives, the LLMs generated navigation sequences that enabled the robots to traverse multiple locations and execute tool pick-and-place operations. The approach of [27] achieved a success rate of up to 92.5%. Additionally, Yin et al. [20] integrated LLM-based task planning with deep reinforcement learning and prompt-based LLM techniques to facilitate task decomposition, allocation, and reward generation. This approach supported human demonstration-assisted robot skill synthesis in HRC assembly tasks, achieving an average task success rate of 95.92%.

Building upon prompt-based direct IO, many studies enhance context reasoning accuracy by incorporating COT prompting and its variants, or by integrating external structured modules. Gkournelos et al. [19] proposed an LLM-based manufacturing execution system that integrates a robot behavior tree mechanism into the planning process for assembly task allocation. This approach enables more structured and interpretable task planning and achieved an average goal accuracy of 93.33%. Gao et al. [21] introduced an HRC-adapted LLM which included an error correction control module for robot action generation. Hua et al. [24] presented an adaptive intelligence

generation method, in which LLM-based planning is enhanced through a dynamic knowledge evolution mechanism for generating assembly solutions. Under this framework, the supported collaboration model, simultaneous collaboration model, and sequential collaboration model achieved task success rates of 91.54%, 89.31%, and 90.78%, respectively. Ranasinghe et al. [25] proposed CogniVera, a dual-agent architecture that employs an internal dialogue mechanism to facilitate HRC in understanding and decomposing assembly tasks. This approach achieved a task completion rate of 96.67%, demonstrating its effectiveness in complex task interpretation and coordination. Izquierdo-Badiola et al. [26] introduced PlanCollabNL, a framework that integrates the structured Planning Domain Definition Language (PDDL) with LLM-based reasoning to mitigate the impact of LLM hallucinations on task allocation. This approach significantly improves performance in complex HRC adaptive plan generation, achieving subgoal-to-favour and subgoal-to-disfavour completion rates of 84% and 86%, respectively.

While traditional scheduling and control methods excel in structured environments with well-defined states, they struggle with the inherent uncertainties and rich contextual cues present in real-world manufacturing settings. VLMs and LLMs, however, demonstrate exceptional capabilities in open-set problems and contextual reasoning within unstructured visual environments, making them well-suited for handling unforeseen or rare anomalies in task planning scenarios that cannot be directly formulated as traditional optimization problems. Although LLM- and VLM-based task planning has achieved notable progress in HRI and smart manufacturing, these approaches still exhibit several limitations. Whether employing direct input-output methods or CoT-based approaches, global planning remains limited to coarse-level understanding; conversely, detailed planning is typically constrained to predefined action primitives specific to a single type of robot. Moreover, even with multimodal reasoning that combines visual and textual inputs, plans generated by LLMs frequently suffer from hallucinations and vague instructions, lacking

the precision and clarity required for reliable robotic execution. Methods that can simultaneously perform global planning while maintaining fine-grained information representation, with clear vision-grounded reasoning and robot executable plans are not yet to be fully explored.

2.2.2 VLM-based HRI for navigation

Vision and Language Navigation (VLN) denotes a robot's capability to execute navigation tasks by integrating visual perception of the environment with the understanding of natural language commands. In this setting, the agent typically receives: (1) an environmental representation, such as images, point clouds, or other 3D perceptual inputs; (2) natural language instructions describing the path toward the goal; and (3) its initial position within the environment. The agent must interpret the instructions, align them with visual inputs, plan an appropriate path, and act accordingly to reach the designated target. By interpreting both visual signals and language instructions, VLMs together with LLMs substantially boost navigation performance in complex VLN scenarios. Based on different navigation scenarios, we categorise these works into indoor navigation, outdoor navigation, and web navigation. The recent representative works using VLMs or LLMs in navigation tasks are summarised in Table 2.2.

Indoor navigation has become a core capability in service robots for household tasks, and it has also been explored in scenarios of human–robot teamwork within industrial environments. Indoor VLN can assist with assembly, help humans retrieve and place tools, manage logistics distribution within factory premises, and inspect and maintain equipment. In VLN, a typical task involves an agent following detailed step-by-step language instructions to move through the environment and reach a designated goal. These instructions are typically clear and specific, describing each step of the action path. Khandelwal et al. [28] explored CLIP's capabilities in embodied AI, discovering that

Table 2.2: VLM-based Navigation Methods

Navigation environment	Method	VLM/LLM	Success rate	Application	Year	Ref
Indoor	EmbCLIP	CLIP	47% (RoboTHOR OBJECTNAV Challenge 2021)	Domestic service robot	2022	[28]
	SHeFU	Switching Image Embedder, Funnel Transformer	83.1% (ALFRED-fc)	Domestic service robot	2023	[29]
	Ego2-Map	CLIP	47% (R2R-CE)	Domestic service robot	2023	[30]
	VLMaps	LSeg, GPT-3.5	62% (custom dataset)	Domestic service robot	2023	[31]
	MiC	GPT-2, CLIP	55.74% (REVERIE)	Remote embodied referring expression, domestic service robot	2023	[32]
	CKR+	CLIP	23.13% (REVERIE)	Remote embodied referring expression, domestic service robot	2024	[33]
	NavGPT	GPT-3.5, BLIP-2	34% (R2R)	Domestic service robot	2024	[34]
	LANA+	CLIP	70.1% (R2R)	Instruction following and generation, domestic service robot	2024	[35]
	ACK	CLIP	59.1% (REVERIE)	Remote embodied referring expression, domestic service robot	2024	[36]
	CONSOLE	ChatGPT, CLIP	72% (R2R)	Domestic service robot	2024	[37]
	DISH	CLIP	44.3% (R2R)	Domestic service robot	2024	[38]
	A vision AI-based HRC assembly approach	GPT-4	\	Human-robot collaboration	2024	[22]
	A vision and language cobot navigation approach	GPT-3.5	\	Human-robot collaboration	2024	[39]
Outdoor	LM-Nav	CLIP, GPT-3	80% (custom dataset)	Urban VLN	2022	[40]
	VELMA	CLIP, GPT-4	23.1% (Map2seq)	Urban VLN	2024	[41]
	VLN-VIDEO	GPT-2	31.7% (Touchdown)	Urban VLN	2024	[42]

CLIP's feature representations encode these primitives more effectively than ImageNet-pretrained backbones. They proposed the EmbCLIP model, which achieved a 47% task success rate in the RoboTHOR OBJECTNAV Challenge 2021. Korekata et al. [29] proposed SheFU, a model composed of switching image embedder and funnel transformer. The proposed model can infer the target object as well as the final destination separately and achieve a task success rate of 83% in ALFRED-fc dataset. Hong et al. [30] presented Ego2-Map based on ViT-B/16 model from CLIP for VLN in continuous environments and could reach a 47% success rate in the R2R-CE dataset.

Apart from simple instruction-following tasks, some works focus on remote embodied referring expression task, which means the instructor only gives high-level instructions, and the robot needs to figure out the clear destination by continuously asking questions according to the current state. Huang et al. [31] presented VLMaps based on LSeg and GPT-3.5, in which they utilised GPT-3.5 to generate Python code for robot navigation, and the method reached a 62% success rate in their custom dataset. Qiao et al. [32] introduced the MiC framework for the remote embodied referring expression task, integrating CLIP for scene interpretation with GPT-2 for in-context learning, and reported a task success rate of 55.74% on the REVERIE benchmark. Similarly, Gao et al. [33] introduced CKR+ based on CLIP which could achieve a 23.13% success rate in REVERIE dataset. Wang et al. [35] devised CLIP-based LANA+ model which could mimic the human process of finding a path through iterative question-and-answer interactions and reached a success rate of 70.1% in R2R dataset. Mohammadi et al. [36] proposed the ACK framework, which leverages a spatio-temporal knowledge graph encoding commonsense information to improve navigation efficiency. In this design, CLIP was applied to extract and rank scene- and object-related knowledge, leading to a 59.1% task success rate on the REVERIE benchmark.

Recent efforts emphasize examining the reasoning potential of LLMs and incorporating it into task planning for VLN scenarios. These studies seek to

leverage the advanced language comprehension and generation capacities of LLMs to enhance decision-making in challenging navigation scenarios. By integrating LLMs with visual information, researchers seek to develop advanced navigation systems capable of interpreting and responding to dynamic scenarios. Zhou et al. [34] developed a purely LLM-based VLN system based on GPT-3.5 and BLIP-2. This work innovatively explored GPT's zero-shot reasoning capabilities in complex environments and achieved a 34% task success rate on the R2R dataset without any training. Similarly, Lin et al. [37] proposed CONSOLE based on ChatGPT and CLIP, and this model gained a 72% success rate in R2R dataset.

Furthermore, VLMs can be leveraged to formulate strategies for robot learning in VLN tasks. This involves using VLMs to generate and optimise action plans that guide robots through various navigation challenges. The adoption of VLMs within robot learning enables improved comprehension of multimodal information while enhancing robustness and efficiency in real environments. Wang et al. [38] introduced a Reinforcement Learning (RL) framework of discovering intrinsic subgoals via hierarchical (DISH) RL in which CLIP's image encoder was applied for visual feature extraction. This method overcame the label annotation problem in reinforcement learning and achieved a 44.3% task success rate in R2R dataset.

While models like EmbCLIP and SheFU demonstrate high task success rates by effectively encoding visual features and predicting target locations, approaches such as VLMaps and MiC emphasize interactive question-asking to improve task completion in more ambiguous scenarios. This highlights a common trend toward integrating visual perception with dynamic linguistic interaction, although the varying success rates underline the ongoing challenge of achieving consistency across different datasets and environments.

In the industrial sector, some works have also introduced LLM-based VLN for HRC tasks. Liu et al. [22] utilised GPT-4 along with 3D object reconstruction

and pose estimation methods in human-robot collaborative assembly tasks, while Wang et al. [39] built a cobot navigation framework using GPT-3.5 combined with other visual methods to assist operators in tool retrieval and placement in HRC. However, while these advancements showcase promising progress, the variability in success rates across different models and datasets highlights a critical challenge in performance consistency, indicating a need for improved generalisation and adaptability in dynamic real-world environments.

Compared to indoor VLN, the outdoor VLN environment is typically more open and expansive, with fewer constraints on movement. Outdoor landscapes can include a variety of terrains, weather conditions, and lighting variations. The presence of dynamic elements such as vehicles and pedestrians adds to the complexity. The applications of outdoor VLN include autonomous driving, robotic delivery, smart city management, rescue operations, and environmental monitoring. In the industrial sector, outdoor VLN can also be employed for factory security patrols and logistics distribution between factory sites. Shah et al. [40] presented a system called LM-Nav based on CLIP and GPT-3 for long-horizon navigation through outdoor and complex environments and achieved an 80% success rate in their custom dataset. Schumann et al. [41] introduced VELMA, which integrates CLIP with GPT-4 to exploit trajectory descriptions and visual scene observations as contextual cues for predicting subsequent actions, achieving 23.10% task success on the Map2seq dataset. Li et al. [42] introduced VLN-VIDEO, utilising urban driving videos combined with automatically generated navigation instructions and actions to enhance outdoor VLN performance. GPT-2 was used to filter out templates with low generation probability. They gained a 31.7% success rate in the Touchdown dataset. Despite these advancements, the significant variation in success rates indicates that outdoor VLN systems must address the challenges of diverse environmental conditions and dynamic elements more effectively to achieve reliable performance in real-world applications.

Based on the above analysis, these methods have two drawbacks. Firstly, these methods are limited to navigating to object landmarks based on 2D top-down map, they did not consider subsequent operations on more fine-grained targets. Secondly, while the currently proposed method works well in the simulator, in the real world, it is affected by sensor noise, which results in the aliasing of geometric detail in the reconstructed maps, thus failing to accurately segment the navigation target.

2.2.3 VLM-based HRI for manipulation

VLMs have also been frequently leveraged in robotic manipulation tasks to enable robots to perform physical tasks by parsing and understanding visual inputs (such as images or videos) and language inputs (such as instructions or descriptions). This type of interaction enables robots to execute more flexible and adaptive operations in complex environments. The methodology in manipulation is similar to navigation; it also requires environmental representation, natural language instructions, and the robot's initial state. VLMs and LLMs can accomplish joint reasoning for visual perception and language comprehension, while also facilitating further trajectory planning. Recent works about VLMs/LLMs in robotic manipulation are provided in Table 2.3.

For vision and language manipulation algorithms, current approaches primarily utilise VLMs for scene understanding and LLMs for natural language command comprehension, combined with planning algorithms to generate trajectory key points or dense waypoints for the end-effector. These methods can be categorised into two types: one involves designing complex prompts for VLMs and LLMs to directly generate trajectories, referred to as planning-based vision and language manipulation, and the other employs VLMs and LLMs to assist in policy generation within robot learning to accomplish manipulation tasks, known as learning-based vision and language manipu-

Table 2.3: VLM-based Manipulation Methods

Method	VLM/LLM	Success rate	Application	Year	Ref
CLIP-SemFeat	CLIP	58.8% (LVIS)	Scene rearrangement, tabletop household tasks	2022	[43]
Act3D	CLIP ResNet50	83% (RLBENCH)	Tabletop household tasks	2023	[44]
CALAMARI	CLIP	84% (RLBENCH)	Contact-rich Manipulation, tabletop household tasks	2023	[45]
GNFactor	Stable Diffusion, CLIP	31.7% (RLBENCH)	Tabletop household tasks	2023	[46]
GVCCI	Grounding Entity Transformer (Get), Setting Entity Transformer (Set)	50.98% (VGPI)	Pick and place task, tabletop household tasks	2023	[47]
MOO	Owl-ViT	∼50% (RT-1)	Tabletop household tasks	2023	[48]
PaLM-E	PaLM-E	66.1% (OK-VQA)	Tabletop household tasks	2023	[49]
SMS	ViLD, BLIP-2, SAM	41.7% (custom dataset)	Mechanical search, tabletop household tasks	2023	[50]
Voxposer	GPT-4, OWL-ViT, SAM	88% (RLBENCH)	Tabletop household tasks	2023	[51]
Vision-language guided robotic planning	CLIP, GPT-4	93.3% (custom dataset)	Human-robot collaboration	2024	[23]

lation. Regarding application scenarios, most work applying VLMs/LLMs to robotic manipulation focuses on tabletop household tasks, with some studies extending their use to industrial settings.

Planning-based vision and language manipulation is an approach in robotic manipulation where VLMs and LLMs are utilised to generate precise movement trajectories for robotic tasks. The technique relies on carefully formulated language instructions, which enable large language models to output detailed motion sequences for the end-effector. Goodwin et al. [43] introduced CLIP-SemFeat, a method for tabletop scene rearrangement that exploits CLIP to address cross-instance matching, achieving a 58.5% success rate on the LVIS dataset. Driess et al. [49] proposed PaLM-E, a vision-language model for robotic manipulation that transfers multimodal knowledge into actionable reasoning, allowing robots to plan under challenging dynamics and constraints while answering queries about their surroundings. PaLM-E could achieve a 66.1% task success rate in OK-VQA dataset. Sharma et al. [50] presented SME consisting of ViLD, BLIP-2 and SAM for mechanic searching for tabletop household tasks, and could reach a 41.7% task success rate in the custom dataset. Huang et al. [51] proposed a novel framework named Voxposer for model-based planning. This framework utilised ViLD for object detection, SAM for semantic segmentation, and GPT-4 for reasoning and planning. VoxPoser achieves waypoint-level accuracy in robotic arm trajectory planning, thereby enhancing the adaptability of manipulation tasks. It achieved an 88% task success rate in the RLBENCH tasks. While these approaches demonstrate significant progress in household settings, the current methods often lack the precision required for complex industrial tasks, underscoring the need for further research to enhance motion planning precision and extend their applicability to industrial environments.

Learning-based vision and language manipulation is an approach where VLMs and LLMs are used to assist in generating policies for robot learning. Rather than producing motion paths explicitly, the approach aims to

learn control strategies that allow robots to acquire manipulation skills by interacting with their surroundings. Gervet et al. [44] proposed Act3D, utilising CLIP ResNet50 and leveraging their shared vision-language feature space to interpret instructions and foundational references for learning-from-demonstration tasks. They achieved an average success rate of 83% in RLBENCH tasks. Wi et al. [45] introduced CALAMARI for contact manipulation in household environments based on CLIP and obtained a success rate of 90% in the wipe_desk task, 84% in the sweep_to_dustpan task, and 60% in the push_button task in their custom industrial dataset. In the industrial sector, VLM-based manipulation, in addition to completing HRC assembly tasks, has potential value in warehouse management, equipment maintenance, flexible production line adjustments, and item sorting, and is worth further exploration.

It is important to note that while planning-based methods like CLIP-SemFeat [43] and Voxposer [51] excel in task success rates for household tasks, they often struggle with the precision required for industrial applications. In contrast, learning-based approaches such as Act3D [44] and CALAMARI [45] show promise in adaptability through learning from interactions but display variability in success rates across different tasks. Identifying these trends and common challenges suggests a potential for integrating the strengths of both approaches to improve precision and adaptability, particularly in transitioning from household to industrial applications. This synthesis highlights a critical trend towards developing hybrid models that can bridge current gaps, offering a more robust solution across varied environments. The adaptation of VLM-based manipulation from household to industrial applications faces challenges such as the need for higher precision, robustness in diverse conditions, and integration with existing industrial systems, indicating significant opportunities for future research to bridge these gaps and fully realise its potential in industrial settings.

Although current VLM-based robot manipulation approaches have made significant progress, they still exhibit several common limitations. Firstly, most of the robot manipulation work in industry predominantly focuses on the execution phase of robotic actions, neglecting the pivotal role of precise visual affordance representation during the perception phase within the entire pipeline of robotic tool-use tasks. This oversight limits the methods' interactivity, intelligence, and capability for precise manipulation. Secondly, there is a scarcity of research that simultaneously addresses the full spectrum of affordances involved in a comprehensive robotic operation pipeline, such as graspable, functional, and movement affordances, rendering it challenging to directly transfer existing work to robotic manipulation tasks. Thirdly, current foundation-model based manipulation approaches possess only rudimentary affordance representations that can handle everyday tasks such as opening drawers to store items or pouring water from a teapot. They are inadequate for industrial tasks requiring precise manipulation.

2.3 Research gaps

The above review of existing literature presents a brief glimpse of the current application status of vision-language-based HRI techniques in HRC environments. Some limitations and research gaps have also been revealed through the review process and are summarized here.

2.3.1 VLM-based HRI for task planning

Firstly, whether using direct IO strategies or CoT reasoning, global planning is still restricted to a coarse-grained understanding of tasks, while detailed planning remains bound to predefined, robot-specific action primitives, often tailored to a single platform.

Secondly, despite progress in multimodal reasoning combining visual and textual inputs, plans generated by LLMs often exhibit hallucinations, ambiguities, and insufficiently detailed instructions, making them unsuitable for reliable robotic execution. Approaches capable of bridging this gap—by uniting global task planning with fine-grained information representation, robust vision-grounded reasoning, and the generation of robot-executable plans—have not yet been fully explored.

2.3.2 VLM-based HRI for navigation

Firstly, learning-based VLN methods heavily rely on data and are constrained by the availability of a fixed number of scenes for training. This limitation prevents them from effectively addressing the challenges of realistic application scenarios, where objects may exist outside the pre-defined domains and language instructions can be more diverse.

Secondly, although the current method shows promising performance in simulated environments, it faces challenges when applied in real-world scenarios due to sensor noise. This noise can lead to the loss of geometric details in the reconstructed maps, resulting in inaccurate segmentation of the navigation target.

2.3.3 VLM-based HRI for manipulation

Firstly, most of the robot manipulation work in industry predominantly focuses on the execution phase of robotic actions, neglecting the pivotal role of precise visual affordance representation during the perception phase within the entire pipeline of robotic tool-use tasks. This oversight limits the methods' interactivity, intelligence, and capability for precise manipulation.

Secondly, there is a scarcity of research that simultaneously addresses the full spectrum of affordances involved in a comprehensive robotic operation pipeline, such as graspable, functional, and movement affordances, rendering it challenging to directly transfer existing work to robotic manipulation tasks.

Thirdly, current foundation-model based manipulation approaches possess only rudimentary affordance representations that can handle everyday tasks such as opening drawers to store items or pouring water from a teapot. They are inadequate for industrial tasks requiring precise manipulation.

3 Task Planning in Unforeseen Scenarios for Human-centric Smart Manufacturing: A Coarse-to-Fine Vision-Grounded Chain-of-Thought Reasoning Approach

3.1 Introduction

Industry 5.0 advocates for manufacturing systems that embody human-centricity, sustainability, and resilience [52]. In response to the growing demand for mass customization, smart manufacturing systems are expected to evolve toward flexible and adaptive operational models that enable close collaboration between humans and manufacturing systems and dynamic adaptation to changing environments, all while keeping human well-being and values at the core of system design. In this case, HRC becomes an indispensable component. It emphasizes the integration of human decision-making and reasoning abilities with the precision and speed of machine task execution, thereby enabling improved production quality and throughput [16]. In this context, task planning plays a crucial role in the whole HRC system, which involves interpreting human instructions and combining them with the current task requirements and environmental conditions to determine the robot's next actions [53, 54].

In most manufacturing processes, task planning is typically regarded as a goal-oriented optimization problem [55, 56]. Existing approaches primarily focus on leveraging complex rule-based methods, optimization algorithms, or reinforcement learning techniques to achieve optimal system performance—such as minimizing makespan, maximizing resource utilization, or balancing workload—in relatively structured and well-defined environments using deterministic information [57, 58, 59]. While traditional task planning methods excel in structured environments with well-defined states, they struggle with the inherent uncertainties and rich contextual cues present in real-world manufacturing settings.

The remarkable evolution of LLMs and VLMs has enabled the extraction of extensive commonsense knowledge from heterogeneous web-based data sources [60]. This accumulated knowledge base presents substantial opportunities for embodied intelligent agents, facilitating the generation of executable task plans that align with human instructions expressed through natural language interfaces and current environments [61]. In this case, these foundation models demonstrate exceptional capabilities in open-set problems and contextual reasoning within unstructured visual environments, making them well-suited for handling unforeseen or rare anomalies in task planning scenarios that cannot be directly formulated as traditional optimization problems [62].

Recent progress in task planning with LLMs and VLMs has substantially advanced HRI and smart manufacturing. The prevailing paradigm leverages prompts containing primitive information that explicitly delineates the desired output content and structure, enabling the model to produce task plans directly [23, 22, 27, 20]. Building on such prompt-based input–output methods, a growing body of research has enhanced contextual reasoning accuracy by employing CoT prompting and related variants. These approaches systematically guide the model through intermediate inference steps, yielding more coherent and structured task plans. In parallel, other studies augment

LLM reasoning with external structured modules—such as knowledge graphs, symbolic planners, or environment simulators—to improve reliability and ensure consistency with real-world constraints [19, 21, 24, 25, 26]. Despite these advances, whether through direct IO methods or CoT-based reasoning, most existing global planning strategies remain limited to high-level abstractions, while fine-grained planning is often restricted to predefined action primitives designed for specific robot platforms. This misalignment impedes the realization of flexible, executable, and interactive planning systems that can adapt to heterogeneous multi-agent environments and dynamic operational contexts. Moreover, even when integrating multimodal reasoning that combines visual and textual information, VLM-generated plans often contain hallucinations and imprecise directives, failing to achieve the accuracy and clarity required for reliable robotic execution.

To fill the gaps, we proposed a coarse-to-fine vision-grounded chain-of-thought reasoning approach for human-robot interactive task planning for unforeseen scenarios. This approach comprises two essential components: multi-granularity visual parsing and coarse-to-fine CoT reasoning. The multi-granularity visual parsing enables hierarchical segmentation and the generation of mark_ids, allowing the VLM to reason effectively based on these identifiers and produce clear operational instructions. The coarse-to-fine CoT framework seamlessly integrates global scene understanding with local perspective analysis, facilitating joint reasoning and the generation of stable, executable task plans.

3.2 Methodology

3.2.1 Problem description

We consider task planning in human-centric smart manufacturing cells operating under unforeseen scenarios, where the environment, resource availability, and human activities may deviate from nominal assumptions without prior enumeration. Let the cell be observed through a visual stream; for a single decision epoch we denote the RGB observation by $I \in \mathbb{R}^{H\times W\times 3}$, and a high-level natural-language goal by $q \in \Sigma^*$. The planner must produce an executable plan

$$P \;=\; \big(\tau_1, \tau_2, \ldots, \tau_T\big), \qquad \tau_t \;=\; \big(a_t,\, o_t,\, r_t\big), \tag{3.1}$$

where $a_t \in \mathcal{A}$ is an atomic skill from a finite library (e.g., pick, place, inspect, dock), o_t is a grounded target (object/region) in the current scene, and r_t denotes resource assignments (tooling, fixtures, AGV/robot endpoints). The plan must satisfy hard constraints capturing human safety, process feasibility, and resource capacities:

$$\mathcal{C}_{\text{safety}}(P, I) \le 0, \quad \mathcal{C}_{\text{proc}}(P) \le 0, \quad \mathcal{C}_{\text{cap}}(P) \le 0, \tag{3.2}$$

including precedence constraints modeled by a partial order $(\mathcal{T}, \prec)$ over subtasks implied by q.

In practice, the environment is partially observed and non-stationary due to human motion and occlusions. We adopt a perception-guided grounding interface that converts I into a finite set of actionable, referable entities via multi-granularity segmentation:

$$\Phi_{\text{c}}(I) \to \mathcal{G}^{\text{c}}, \qquad \Phi_{\text{f}}(I, \mathcal{R}) \to \mathcal{G}^{\text{f}}(\mathcal{R}), \tag{3.3}$$

where Φ_{c} yields coarse instance/region proposals $\mathcal{G}^{\mathrm{c}}$, $\mathcal{R} \subseteq \mathcal{G}^{\mathrm{c}}$ is a selected subset of regions of interest, and Φ_{f} refines within $\mathcal{R}$ to produce fine-level segments $\mathcal{G}^{\mathrm{f}}(\mathcal{R})$. Each segment $g \in \mathcal{G}^{\mathrm{c}} \cup \mathcal{G}^{\mathrm{f}}(\mathcal{R})$ is assigned a unique marker $m(g)$ that supports unambiguous linguistic reference and downstream action binding.

Because exhaustive plan search is combinatorial—$|\mathcal{A}|^T$ in the worst case—and grounding under shift is brittle, we employ a coarse-to-fine chain-of-thought (CoT) reasoning paradigm to structure the plan search space. Concretely, we define selection rules such as availability and accessibility for risk-aware plan selection

$$F_\theta\Big(P \,\Big|\, q,\, I,\, \mathcal{M}\Big) \tag{3.4}$$

where $\mathcal{M} = \{m(g)\}$ is the set of visual markers and F_θ is induced by a vision-language reasoner prompted to produce explicit intermediate rationales. The desired solution is then

$$P^\star \;\in\; \arg\max_{P \in \mathcal{F}(I,q)} F_\theta\Big(P \,\Big|\, q,\, I,\, \mathcal{M}\Big), \tag{3.5}$$

computed via a two-stage CoT process: (i) coarse planning that selects and orders macro-subtasks over $\mathcal{G}^{\mathrm{c}}$ to satisfy precedence and safety, and (ii) fine planning that resolves object/tool choices and motion-level bindings using $\mathcal{G}^{\mathrm{f}}(\mathcal{R})$. This formulation captures the core requirement: generate safe, feasible, and robust plans that adapt to unforeseen visual and operational variations by iteratively narrowing the hypothesis space with perception-anchored, language-mediated reasoning.

3.2.2 Framework

We proposed coarse-to-fine, vision-grounded CoT reasoning framework that maps the perceptual input I, goal q, and optional context c into an executable plan P, as shown in Fig. 3.1. The framework consists of two critical compo-

nents: (i) multi-granularity segmentation and numeric marker generation, (ii) vision-grounded CoT prompting for coarse situational reasoning with region-of-interest (ROI) selection, and fine task reasoning with grounded actuation binding. Both coarse and fine reasoning operate strictly on segmented and marked images to ensure unambiguous visual grounding: the coarse stage uses a global view, and the fine stage uses localized, detail-rich views targeted to the planned object of interaction.

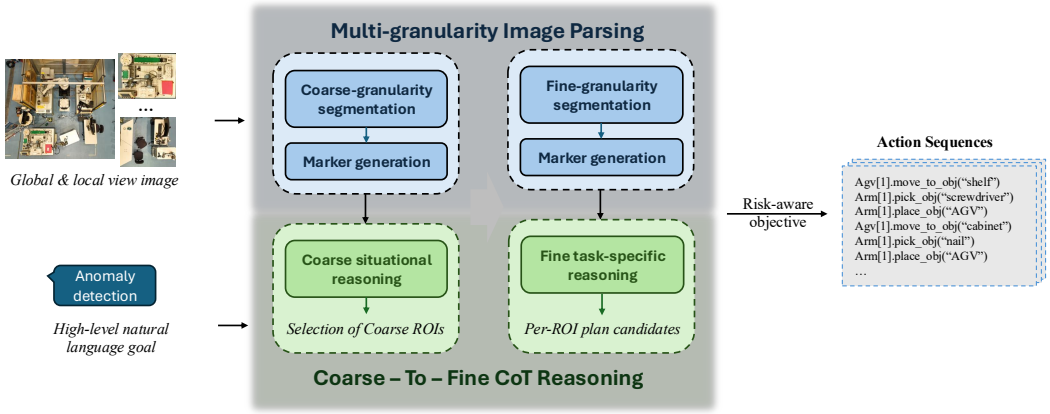


Fig. 3.1: Framework of coarse-to-fine vision-grounded chain-of-thought reasoning approach.

Formally, the perception component produces discrete, referable entities at two granularities:

$$\Phi_{\mathrm{c}}(I) \;=\; \Big(\mathcal{G}^{\mathrm{c}},\, \mathcal{M},\, \mathbf{M}^{(\mathrm{c})}\Big), \qquad \Omega_{\mathrm{c}} \;=\; \Omega\Big(I, \mathbf{M}^{(\mathrm{c})}\Big), \tag{3.6}$$

where $\mathcal{G}^{\mathrm{c}}$ are coarse segments, $\mathcal{M} = \{m(g)\}$ are numeric markers, $\mathbf{M}^{(\mathrm{c})}$ are binary masks, and $\Omega(\cdot)$ renders edges and labels on the image (cf. Section 3). Coarse CoT prompting consumes only the marked global image:

$$\begin{aligned} \kappa_{\mathrm{c}} \;&=\; f_{\mathrm{c}}\Big(q_{\mathrm{c}},\, \Omega_{\mathrm{c}}\Big) \\ \equiv\; &\Big(\texttt{priority\_assessment}, \texttt{resource\_distribution}, \dots\Big), \end{aligned} \tag{3.7}$$

followed by deterministic selection of coarse ROIs via

$$\mathcal{R} \;=\; \mathsf{Sel}\Big(\texttt{roi\_ids},\, \mathcal{G}^{\mathrm{c}}\Big), \tag{3.8}$$

For each $g \in \mathcal{R}$, we collected the fine, task-aligned view image I'_g according to different tasks. For navigation, I'_g centers on the relevant lane/zone region, whereas for manipulation it centers on the target workstation/tool workspace. Fine-granularity segmentation and marking then yield

$$\Phi_{\mathrm{f}}\Big(I'_g\Big) \;=\; \Big(\mathcal{G}^{\mathrm{f}}(g),\, \mathcal{M}'(g),\, \mathbf{M}^{(\mathrm{f})}(g)\Big), \tag{3.9}$$

$$\Omega_{\mathrm{f}}(g) \;=\; \Omega\Big(I'_g,\, \mathbf{M}^{(\mathrm{f})}(g)\Big). \tag{3.10}$$

Fine CoT prompting is performed exclusively on the marked fine view with coarse context:

$$h(g) \;=\; f_{\mathrm{f}}\Big(q_{\mathrm{f}}^{(r,t)},\, \Omega_{\mathrm{f}}(g),\, \psi(\kappa_{\mathrm{c}}),\, m(g)\Big), \tag{3.11}$$

where $\psi(\kappa_{\mathrm{c}})$ is the coarse summary string used for conditioning, and $h(g)$ is a JSON-structured hypothesis containing at least the action and target marker. Aggregating per-ROI hypotheses forms a finite candidate set $\mathcal{P}_{\mathrm{cand}}$, sequenced according to `commands` and the precedence relation $(\mathcal{T}, \prec)$ implied by (q, c).

Finally, we select the executable plan using the risk-aware objective defined in formula 3.4. This two-component design enforces a strict grounding contract: every coarse and fine reasoning step operates on the corresponding parsed image Ω_{c} or $\Omega_{\mathrm{f}}(g)$, ensuring that all references in P are discrete, numeric, and traceable to pixel-level supports, while task-aware fine views supply the detail necessary for robust action binding in unforeseen scenarios.

3.2.3 Multi-granularity visual parsing

Multi-granularity visual parsing, derived from SoM [63], enables images to be segmented and annotated at multiple levels of granularity. This hierarchical representation provides richer structural information, which in turn supports and enhances the reasoning capabilities of VLMs. We implement the perception interfaces Φ_c and Φ_f through a hierarchical neural parsing pipeline that converts RGB observations into discrete, numerically-indexed object references, as shown in Fig. 3.2.

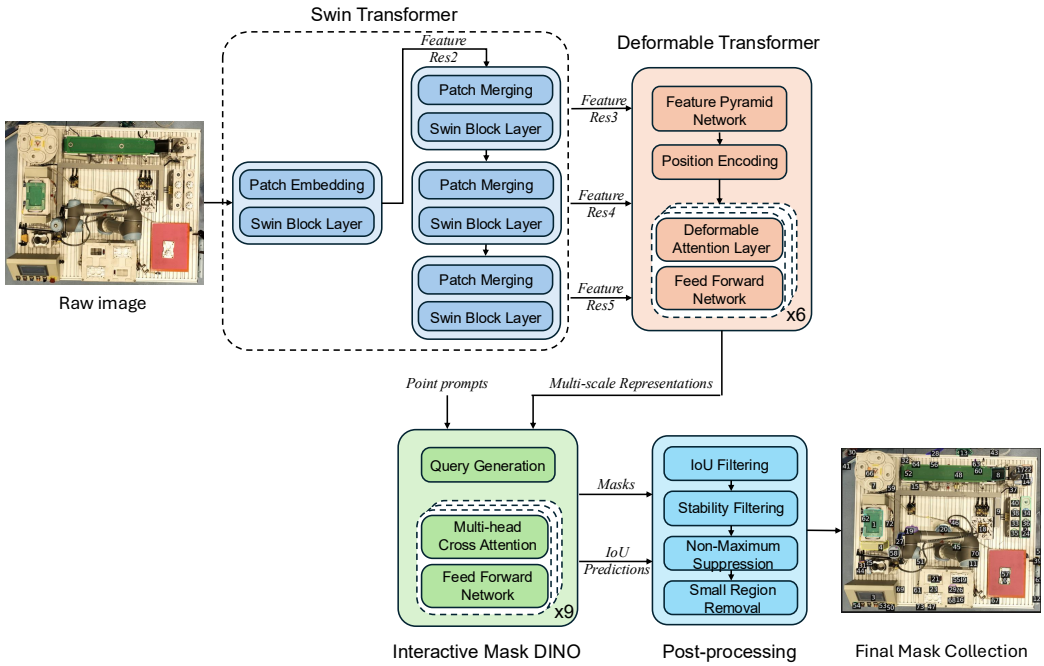


Fig. 3.2: Multi-granularity visual parsing [63].

The core architecture employs a Swin Transformer backbone [64] followed by a deformable attention encoder [65] and an Interactive Mask DINO decoder [66], enabling controllable granularity through slider-based model selection and level-conditioned inference.

Backbone feature extraction. Given an input image $I \in \mathbb{R}^{H\times W\times 3}$, we first apply patch embedding with 4×4 patches to obtain initial tokens of

dimension 192. The Swin Transformer backbone processes these through four hierarchical stages:

$$
\begin{aligned}
&\text{S1:} \quad \text{SwinBlock}^{(1)}(d=2, h=6) \rightarrow \text{res2} \in \mathbb{R}^{H/4 \times W/4 \times 192} \\
&\text{S2:} \quad \text{PatchMerge} + \text{SwinBlock}^{(2)}(d=2, h=12) \rightarrow \text{res3} \in \mathbb{R}^{H/8 \times W/8 \times 384} \\
&\text{S3:} \quad \text{PatchMerge} + \text{SwinBlock}^{(3)}(d=18, h=24) \rightarrow \text{res4} \in \mathbb{R}^{H/16 \times W/16 \times 768} \\
&\text{S4:} \quad \text{PatchMerge} + \text{SwinBlock}^{(4)}(d=2, h=48) \rightarrow \text{res5} \in \mathbb{R}^{H/32 \times W/32 \times 1536}
\end{aligned}
\tag{3.12}
$$

where d denotes layer depth, h the number of attention heads, and each stage employs 12×12 sliding windows for efficient local attention computation.

Deformable attention enhancement. The multi-resolution features $\{\text{res3}, \text{res4}, \text{res5}\}$ are passed through a Feature Pyramid Network [67] to align dimensions to 256, followed by 2D positional encoding. Six deformable transformer encoder layers then produce enhanced representations:

$$
\mathbf{F}^{(\ell+1)} = \text{FFN}\Big(\mathbf{F}^{(\ell)} + \text{MSDeformAttn}(\mathbf{F}^{(\ell)}; 8 \text{ heads}, 4 \text{ levels}, 4 \text{ points})\Big), \quad \ell = 0, \ldots, 5 \tag{3.13}
$$

where $\mathbf{F}^{(0)}$ represents the FPN-aligned features and each deformable attention layer aggregates information across spatial locations and scale levels with learnable offsets.

Interactive mask generation. Point prompts $\mathbf{P} \in \mathbb{R}^{N \times 4}$ (with coordinates and scale factors) are converted to learnable object queries. These queries undergo nine transformer decoder layers, each containing multi-head

self-attention, cross-attention with enhanced features $\mathbf{F}^{(6)}$, and feed-forward processing:

$$\begin{aligned}\mathbf{Q}^{(\ell+1)} &= \text{MHA}(\mathbf{Q}^{(\ell)}) + \mathbf{Q}^{(\ell)} \\ \mathbf{Q}^{(\ell+1)} &= \text{CrossAttn}(\mathbf{Q}^{(\ell+1)}, \mathbf{F}^{(6)}) + \mathbf{Q}^{(\ell+1)} \\ \mathbf{Q}^{(\ell+1)} &= \text{FFN}(\mathbf{Q}^{(\ell+1)}) + \mathbf{Q}^{(\ell+1)}, \quad \ell = 0, \ldots, 8.\end{aligned} \tag{3.14}$$

The final queries are passed to dual prediction heads: a mask MLP generating logits $\mathbf{M}_{\text{raw}} \in \mathbb{R}^{N \times H \times W}$ and an IoU head producing quality scores $\mathbf{S}_{\text{IoU}} \in \mathbb{R}^{N}$.

Then the granularity control is applied via a level parameter $\ell \in \{1, 2, 3, 4, 5, 6\}$ that controls instance granularity within Semantic-SAM.

Post-processing. Raw masks undergo a cascaded quality control pipeline: IoU filtering ($\mathbf{S}_{\text{IoU}} > 0.88$), stability scoring ($\text{stability}(\mathbf{M}) \geq 0.92$), non-maximum suppression (IoU threshold 0.7), and small region removal (area < 10 pixels). Surviving masks are sorted by area in descending order and assigned unique integer markers:

$$m(g_i) = i, \quad i = 1, \ldots, K, \tag{3.15}$$

where $K \leq K_{\text{c}}$ is the number of quality-filtered segments. Geometric descriptors are computed as

$$\mu_i = \frac{1}{|g_i|} \sum_{(x,y) \in g_i} \begin{bmatrix} x \\ y \end{bmatrix}, \quad B_i = \text{bbox}(g_i) \oplus \Delta, \tag{3.16}$$

supporting centroid-based marker placement and ROI-conditioned cropping for fine-stage analysis.

To ensure unambiguous reference, we render edge outlines and numeric labels through

$$E(M_i) = \text{Dilate}(M_i, k = \max(2, \lfloor \min(H, W)/256 \rfloor + 2)) \setminus M_i, \tag{3.17}$$

where morphological dilation creates bold boundaries, and integer labels $m(g_i)$ are overlaid at centroids μ_i using adaptive font sizing. The marked image is

$$\Omega_{\mathrm{c}} = \Omega(I, \mathbf{M}^{(\mathrm{c})}) = \text{Composite}(I, \{E(M_i)\}, \{\mu_i\}, \{m(g_i)\}), \tag{3.18}$$

providing the discrete reference interface required by downstream CoT prompting. Optional category-aware highlighting $\mathrm{Hi}(I, \mathbf{M})$ can be applied without altering $\mathcal{G}^{\mathrm{c}}$ to emphasize safety-critical assets (workstations, mobile robots) via single-shot VLM classification and selective edge rendering.

The complete coarse parsing yields $\Phi_{\mathrm{c}}(I) = (\mathcal{G}^{\mathrm{c}}, \mathcal{M}, \mathbf{M}^{(\mathrm{c})})$ where $\mathcal{M} = \{1, \ldots, K\}$ are numeric markers, and the fine parsing $\Phi_{\mathrm{f}}(I', \mathcal{R}) = (\mathcal{G}^{\mathrm{f}}(\mathcal{R}), \mathcal{M}', \mathbf{M}^{(\mathrm{f})})$ operates on task-aligned crops $I' = \rho_{r,t}(I, g, c)$ to provide high-resolution detail. This ensures that every object reference in the candidate plan set $\mathcal{P}_{\mathrm{cand}}$ corresponds to a numerically-indexed, pixel-grounded region, enabling precise actuation binding while maintaining referential consistency across the coarse-to-fine reasoning pipeline.

3.2.4 Coarse-to-fine CoT prompting

We operationalize the scoring functional F_θ with a vision-language reasoner that is constrained to produce JSON-structured outputs grounded in the discrete marker space $\mathcal{M} \cup \mathcal{M}'$. The prompting scheme follows a coarse-to-fine CoT contract in which intermediate rationales are elicited implicitly to guide selection over $\mathcal{P}_{\mathrm{cand}}$, while the emitted API output remains machine-readable and action-ready, as shown in Fig. 3.3.

We define two prompt families:

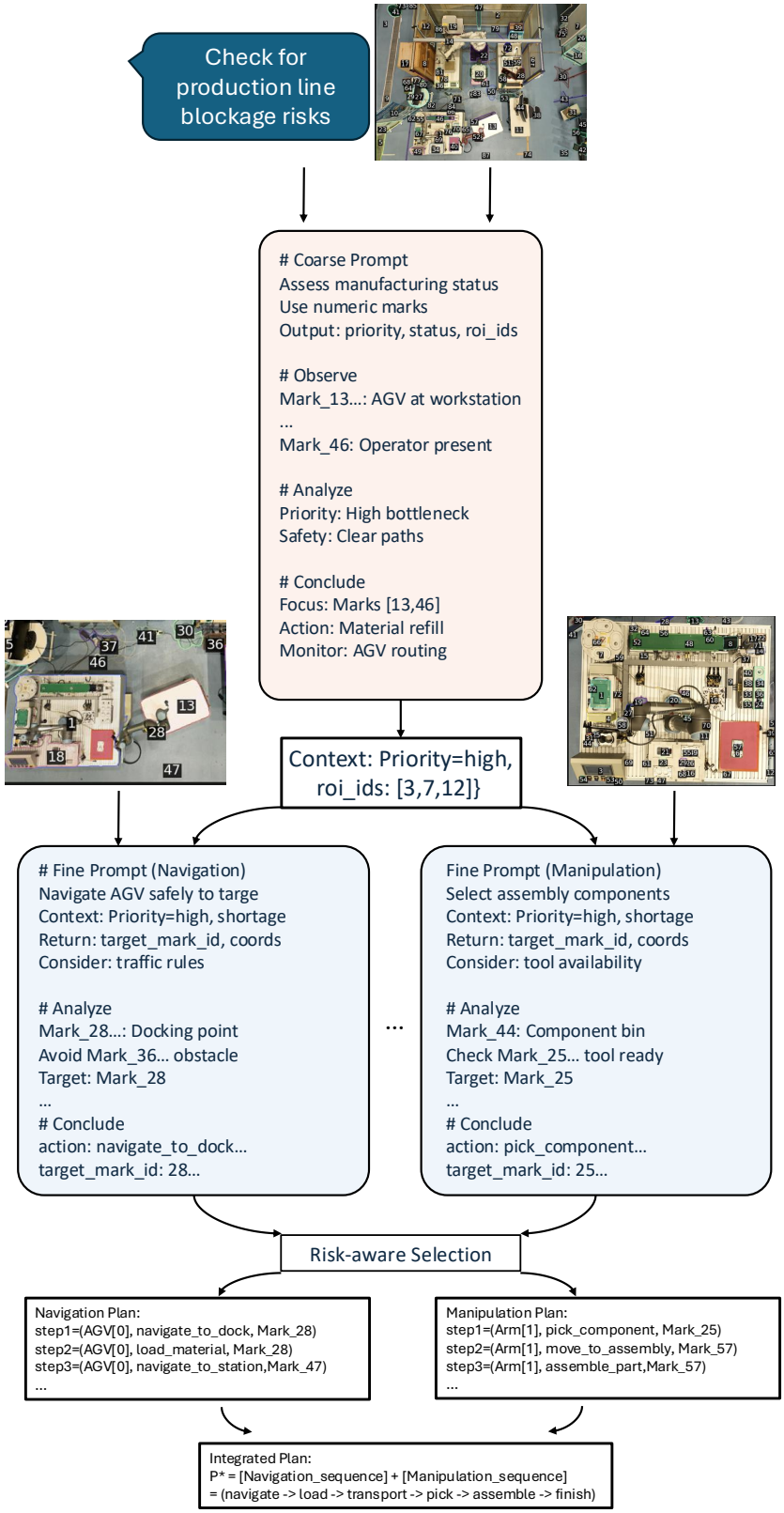


Fig. 3.3: Coarse-to-fine CoT prompting.

$$
\begin{aligned}
\text{Coarse:}\quad & f_{\mathrm{c}}(q_{\mathrm{c}}, I, \mathcal{M}) \Rightarrow \kappa_{\mathrm{c}} \;\equiv\; \big(\texttt{priority\_assessment}, \\
& \qquad \texttt{resource\_distribution}, \texttt{commands}, \texttt{roi\_ids}, \dots\big), \\
\text{Fine:}\quad & f_{\mathrm{f}}\Big(q_{\mathrm{f}}^{(r,t)}, I', \mathcal{M}', \psi(\kappa_{\mathrm{c}}), m(g)\Big) \Rightarrow h \;\equiv\; \big(\texttt{action}, \\
& \qquad \texttt{target\_mark\_id}, \texttt{parameters}, \dots\big),
\end{aligned}
\tag{3.19}
$$

where q_{c} and $q_{\mathrm{f}}^{(r,t)}$ are task-conditioned natural-language templates selected from a curated library for coarse "global monitoring" and fine "navigation/manipulation" subtasks, respectively; $r \in \{\text{navigation}, \text{manipulation}\}$, task t is one of the supported modes (intersection, docking, zone, assembly, picking, tool). The coarse-stage summary κ_{c} is deterministically compressed into $\psi(\kappa_{\mathrm{c}})$ and injected as conditioning for the fine prompts.

Each prompt is paired with parsed and marked images to enforce referential grounding. Let $\Omega(I, \mathbf{M})$ denote the rendered image with numeric labels and outlines. The VLM input at the coarse stage is

$$
U_{\mathrm{c}} \;=\; \Big(S_{\mathrm{sys}},\; q_{\mathrm{c}} \;\oplus\; \text{“Use numeric marks when referring to regions.”},\; \Omega(I, \mathbf{M})\Big),
\tag{3.20}
$$

and at the fine stage for ROI $g \in \mathcal{R}$,

$$
U_{\mathrm{f}}(g) \;=\; \Big(S_{\mathrm{sys}},\; \psi(\kappa_{\mathrm{c}}) \;\oplus\; q_{\mathrm{f}}^{(r,t)} \;\oplus\; \text{“Return ’target\_mark\_id’.”},\; \Omega(I', \mathbf{M}')\Big).
\tag{3.21}
$$

The system instruction S_{sys} enforces JSON-only responses and reiterates the numeric-marker convention. Low sampling temperature ($\tau \approx 0.1$) is used to stabilize decoding under distribution shift.

Coarse reasoning proposes $\mathcal{R} = \mathsf{Sel}(\kappa_{\mathrm{c}})$ with indices $\texttt{roi\_ids} \subseteq \mathcal{M}$. Fine prompting then iterates over ROIs to form hypotheses

$$\mathcal{H} \;=\; \bigcup_{g \in \mathcal{R}} f_{\mathrm{f}}\Big(q_{\mathrm{f}}^{(r,t)},\, I'_g,\, \mathcal{M}',\, \psi(\kappa_{\mathrm{c}}),\, m(g)\Big)\,, \tag{3.22}$$

where each $h \in \mathcal{H}$ specifies a grounded skill instance $\tau = (a, o, r)$ with object/region binding

$$o \;\equiv\; \Big(m(g),\, \mathbf{p}(g)\Big), \qquad \mathbf{p}(g) \in \mathbb{R}^2 \text{ inside the mask of } g. \tag{3.23}$$

If $\mathcal{R} = \varnothing$, a single fine hypothesis is produced on $\Omega(I', \mathbf{M}')$ with the same output contract.

To align with constraints $\mathcal{F}(I, q)$, prompts include explicit guardrails: (i) prioritize human safety and traffic rules in navigation, (ii) respect tool/workstation availability in manipulation, and (iii) surface abnormal workstation states in $\psi(\kappa_{\mathrm{c}})$. These cues bias $p_\theta(Z \mid U)$ toward rationales that reduce the risk penalty $\mathsf{Risk}(P; I, q)$, operationalized via the instruction design and the ROI highlighting prior. Finally the plan selection objective is evaluated by task risk, as shown in formula 3.4. The numeric-marker contract ensures that every action reference in P is unambiguous and traceable back to a pixel-level support, closing the loop between perception, CoT prompting, and risk-aware plan selection.

3.3 Experiments

3.3.1 Multi-granularity visual parsing

The role of visual parsing lies in task-driven segmentation and annotation of visual observations from both the global workshop perspective and the robot's local viewpoint. It serves as the foundation of the entire methodolog-

ical pipeline and is critical for subsequent vision-grounded CoT reasoning. Given that this work specifically targets reasoning in unforeseen scenarios and conditions, the experiments primarily focus on evaluating the model's reasoning performance from both global and local perspectives in previously unseen visual observations.

Experiment settings

To evaluate the model's accuracy, we collected images from the web depicting global views of factory workshops along with corresponding local views of workstations or transport areas. We then applied data augmentation techniques, including brightness adjustment and image flipping. A zero-shot test corpus comprising 910 images was constructed to evaluate the model's ability to reason in scenarios not encountered during training.

The primary evaluation metric for assessing the model's reasoning accuracy is its ability to correctly generate the task-related target IDs. Additional metrics include reasoning time and consistency across multiple reasoning attempts.

In current smart manufacturing tasks, although industrial robots may vary widely in form and function depending on the task, production line setup, and environmental complexity, their core functionalities can generally be categorized into two types: manipulation and navigation, corresponding respectively to robotic arms and mobile robots. For coarse reasoning from the global perspective, the primary objective is to identify the ROI, which is essential for subsequently invoking the corresponding local view for fine-grained reasoning. Therefore, in the global-view reasoning stage, we focus on identifying key targets within the manipulation zone, specifically robotic arms, and within the navigation zone, namely mobile robots. Regarding the visual parsing of the local view, its primary function is to facilitate the generation of executable plans for each specific robot. Therefore, the inference should focus on identifying parts on the workbench and navigable areas as navigation

targets. In addition, we defined five levels of complexity based on the number of targets present in the scene, and evaluated the inference results of both global and local view visual parsing for each complexity level.

Global-view visual parsing

The overall inference performance for ROIs located in the robot arm workstation and the mobile robot platform is presented in Table3.1.

Table 3.1: Global view parsing performance

ROI	Precision	Recall	F1-score	Consistency	Inference time
Robot arm workstation	64.51%	56.92%	59.65%	90.09%	3.42s
Mobile robot platform	74.35%	61.11%	65.47%	97.44%	3.81s

As shown in Table3.1, in the global view parsing task, the inference accuracy for ROIs in the Mobile Robot Zone is generally higher than that for ROIs in the Robot Arm Zone. However, both categories achieve accuracy levels above 50%, which is quite remarkable in a zero-shot evaluation setting. The discrepancy between the two types of ROIs primarily stems from differences in the factory environment. The space in the robot arm manipulation zone is typically more confined and cluttered, making visual parsing more challenging. In contrast, the mobile robot platform zone is usually more spacious to allow for robot navigation, providing clearer visual cues for the model. The model's consistency is measured by calculating the correlation between two inference passes. The results show that both types of ROIs achieve over 90% consistency, indicating that the model demonstrates robust performance across repeated inferences.

Tables3.2 and Table3.3 provide a more detailed breakdown of the performance of visual parsing models under varying levels of scene complexity. In this context, a higher level indicates a greater number of targets present in the observation, corresponding to increased scene complexity.

Table 3.2: Global view parsing - robot arm workstation

Complexity level	Precision	Recall	F1-score
level1	83.33%	75.00%	78.95%
level2	65.38%	56.67%	60.71%
level3	53.33%	60.00%	56.47%
level4	62.50%	50.00%	55.55%
level5	75.00%	30.00%	42.86%

Table 3.3: Global view parsing - mobile robot platform

Complexity level	Precision	Recall	F1-score
level1	73.91%	85.00%	79.07%
level2	76.92%	66.67%	71.43%
level3	53.66%	55.00%	54.32%
level4	82.85%	58.00%	68.26%
level5	100.00%	25.00%	40.00%

As shown in Tables3.2 and Table3.3, as the scene complexity increases, the model's inference precision remains relatively stable. This indicates that the model does not tend to misidentify ROI information even in more complex scenes. However, recall drops significantly with increasing complexity, suggesting that the model tends to miss more ROI information in such scenarios, which in turn leads to a decline in overall inference performance.

Local view visual parsing

Unlike the global view, the local view is based on predefined ROIs and involves parsing local observations obtained from specific robots or regions. Compared to global view observations, local views contain more detailed information, which increases the complexity and difficulty of the inference process.

Table 3.4: Local view parsing performance

ROI	Precision	Recall	F1-score	Consistency	Inference time
Manipulation zone	55.80%	27.27%	36.28%	75.43%	3.62s
Navigation zone	54.92%	35.00%	41.49%	95.97%	4.46s

Table3.4 presents the parsing performance under the local view setting, based on observations collected separately from the manipulation zone and the navigation zone. Compared to the performance of the global view, the local view exhibits slightly lower results due to its higher sensitivity to fine-grained details. The consistency in the navigation zone remains relatively high, exceeding 90%; however, in the manipulation zone, consistency drops to 75.43%, indicating a decline in the model's robustness when parsing fine-grained details. Nonetheless, the inference time remains approximately constant at around 4 seconds.

Table 3.5: Local view parsing - manipulation zone

Complexity level	Precision	Recall	F1 score
level1	64.71%	36.67%	46.81%
level2	47.06%	26.67%	34.04%
level3	46.15%	20.00%	27.91%
level4	40.00%	20.00%	26.67%
level5	100.00%	30.00%	46.15%

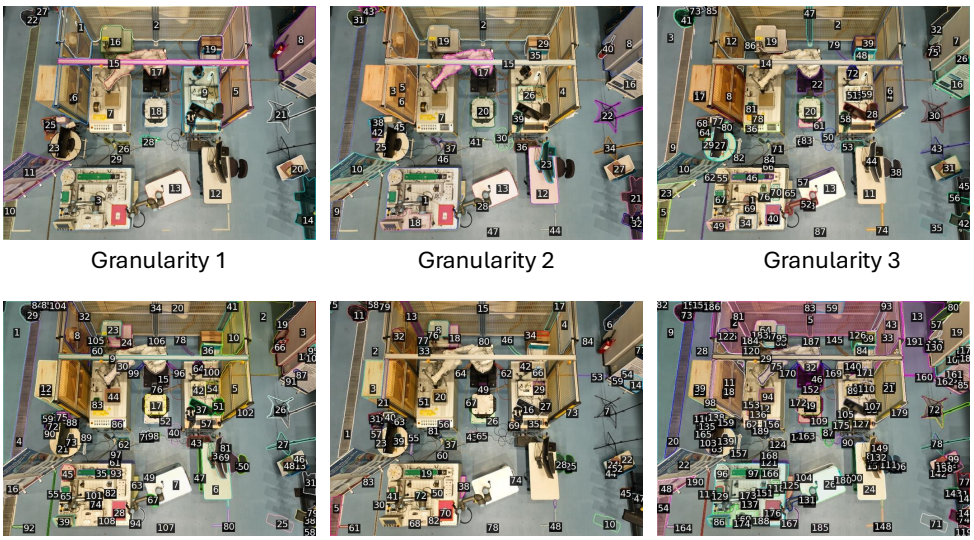
Table 3.6: Local view parsing - navigation zone

Complexity level	Precision	Recall	F1 score
level1	75.00%	60.00%	66.67%
level2	50.00%	16.67%	25.00%
level3	42.86%	22.50%	29.51%
level4	47.62%	50.00%	48.79%
level5	63.64%	35.00%	45.16%

Table3.5 and Table3.6 illustrate that as complexity increases, precision remains relatively stable while recall declines significantly. This trend is consistent with that observed in the global view, suggesting that increased complexity does not lead to a higher rate of false positives, but rather results in more missed detections, thereby diminishing the overall inference performance.

Fig. 3.4 presents the visualization results of multi-granularity image parsing. In Fig. 3.4(a), six different parsing granularities of the global view are displayed. The figure illustrates that the global perspective encompasses

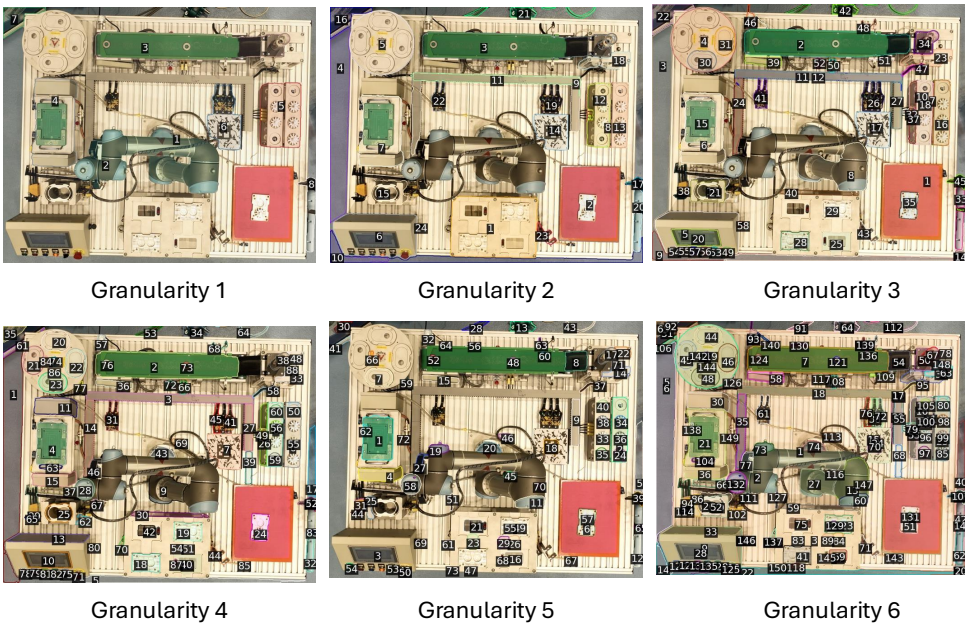
(a) Multi-granularity image parsing for global view



Granularity 1 Granularity 2 Granularity 3

Granularity 4 Granularity 5 Granularity 6

(b) Multi-granularity image parsing for local view



Granularity 1 Granularity 2 Granularity 3

Granularity 4 Granularity 5 Granularity 6

Fig. 3.4: Visualization of multi-granularity in global and local views.

abundant information: granularities 1 and 2 are overly sparse and fail to capture critical details, whereas granularity 6 is excessively dense and introduces redundant information. Consequently, granularities 4 and 5 provide the most appropriate fine-grained representations for this perspective. Fig. 3.4(b) depicts the parsing results of six granularities for the local view. Because the local perspective contains relatively limited content with higher resolution, granularity 6 proves most effective, as it retains all essential information and enables clearer inference. The proposed multi-granularity visual parsing approach enables the model to adapt to diverse visual perspectives and accurately capture the most informative features, thereby facilitating and enhancing coarse-to-fine cot prompting.

3.3.2 Coarse-to-fine CoT prompting

A series of ablation studies were undertaken to examine the role of each component within the marked image-based vision-grounded CoT approach. Evaluations covered several dimensions such as accuracy of outputs, rate of completion, and computational latency.

Experiment settings

In this work, 10 different manufacturing scenarios and tasks were designed to evaluate the method's capabilities in both coarse and fine-grained reasoning. GPT-4o [68] was selected as the VLM to perform reasoning on the input marked image. The evaluation metrics include: fields completion rate, precision, constraint compliance rate, response length, and inference time. Fields completion rate is the proportion of outputs containing all condition-specific required JSON fields. Precision calculated as a normalized spatial accuracy score tied to task stringency and grounding. Constraint compliance rate means the fraction of responses that evidence adherence to at least half of the specified rule categories via rule–keyword matches (e.g., safety, precision,

identification, alignment). Response Length is the character count of the returned content. Also, the end-to-end latency per query is considered as the inference time. Table3.7 presents the average performance of the three types of methods across 10 experimental runs.

Table 3.7: Ablation study

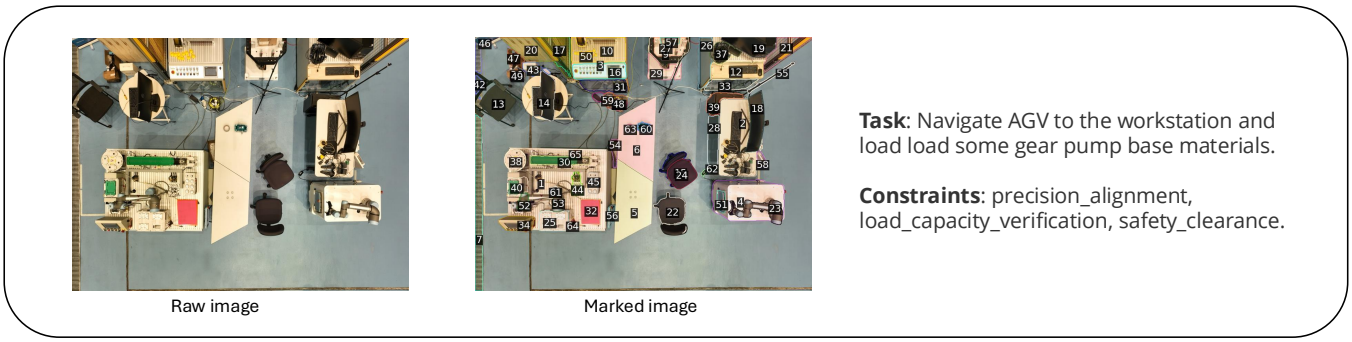
Method	Fields complete rate	Precision	Constraint compliance rate	Response length	Inference time
Raw image + Decision making	0.60	0.10	0.00	115.2	7.05s
Raw image + CoT + Decision making	0.60	0.12	0.10	1041.5	15.17s
Marked image + CoT + Decision making	0.90	0.60	0.80	1244.0	15.64s

Ablation study

As shown in Table3.7, the ablation study demonstrates that reasoning based on the marked image combined with COT prompting substantially enhances performance across multiple critical metrics. Incorporating visual indicators markedly enhances both accuracy and success rates in task execution, indicating that explicit visual cues are fundamental for the model's information processing and handling of complex activities. Moreover, the evident rise in adherence to constraints demonstrates that this approach substantially enhances the model's ability to operate within predefined boundaries, a factor essential for dependable use in structured settings. However, these performance gains come at a computational cost: the additional processing required for CoT-based reasoning results in a near doubling of inference time compared to simpler decision-making approaches. The substantial expansion in response length further suggests that CoT enables more comprehensive and explicit reasoning pathways, albeit with increased computational overhead. These findings collectively underscore the value of multimodal reasoning incorporating both visual markers and structured thought processes, while also

emphasizing the need for optimized implementations to balance accuracy with operational efficiency in real-time applications.

Input



Output

Raw image + VLM decision making


```
{
  "observation": "Industrial workspace with robotic arms and computer stations.",
  "action": "Navigate AGV to the automated loading bay.",
  "target": "Automated loading bay"
}
```


Raw image + CoT reasoning + VLM decision making


```
{
  "observation": "The scene includes several workstations with computers, robotic arms, chairs, and various equipment. There are cables on the floor and a clear path between the workstations.",

  "challenges": "Navigating around obstacles such as chairs and cables, ensuring precise alignment with the loading bay, and verifying load capacity.",

  "constraint_analysis": "Precision alignment requires careful navigation and docking. Obstacle avoidance is crucial due to the presence of chairs and cables. Load capacity verification ensures the AGV can handle the materials. Safety clearance is necessary to prevent collisions.",

  "reasoning": "1. Use sensors to map the environment and identify obstacles. 2. Plan a path avoiding chairs and cables. 3. Approach the loading bay slowly for precision alignment. 4. Verify load capacity before docking. 5. Ensure safety clearance throughout the operation.",

  "action": "Navigate to the loading bay and perform precision docking.",

  "target": "Automated loading bay near the robotic arms.",

  "rationale": "The area near the robotic arms is likely the loading bay. Ensure precision alignment and obstacle avoidance."
}
```


Marked image + CoT reasoning + VLM decision making


```
{
  "mark_inventory": {
    "1": "Workstation", "25": "Materials", "7": "Pathway",
    "4": "AGV", "22": "Chair", "24": "Chair"...
  },

  "relevant_marks": [4, 7, 22, 24, 25],

  "constraint_compliance": {
    "obstacle_avoidance": "Avoid mark 22 and mark 26 while navigating.",
    "load_capacity_verification": "Verify at mark 25 before docking.",
    "safety_clearance": "Ensure clearance at mark 64."
  },

  "spatial_reasoning": "AGV at mark 4 should navigate through mark 7, avoiding mark 22 and 24, to reach mark 25 for material loading.",

  "detailed_plan": "1. Start AGV at mark 4. 2. Navigate through pathway ar mark 7. 3. Avoid obstacle at mark 22, 24. 4. Verify load capacity at mark 25. 5. Align with loading bay at mark 25. 6. Dock near mark 25.",

  "action": "Navigate and dock",
  "target_mark_id": 25
}
```


Fig. 3.5: Demo of ablation experiment results.

Fig. 3.5 illustrates the impact of CoT reasoning and the use of marked images on inference outcomes for the same image and identical task. If the VLM is prompted to reason directly over the raw image, the output tends to be very brief and generic, which makes the commands unusable for robotic execution, as they lack actionable detail. In contrast, when CoT reasoning is applied on the raw image, the generated content contains richer

details and more precise reasoning. However, even with detailed language describing both the action and target, this information remains ambiguous from the robot's perspective, preventing it from executing the plan reliably. By comparison, when CoT reasoning is performed on the marked image, the VLM can directly reference specific targets through their mark IDs. This leads to more concise and efficient reasoning, while the produced commands are explicit and unambiguous. Combined with subsequent coordinate calibration, the robot can unerringly identify both the action and the target, and thus complete the given task accurately.

3.3.3 Case study

Experiment settings

A case study carried out in an actual manufacturing setting is presented to demonstrate the practicality and importance of the proposed approach in smart manufacturing contexts. As shown in Fig. 3.6(1), the entire setup consists of a workstation equipped with a robotic arm, a mobile platform carrying another robotic arm, and a variety of manufacturing-related parts. In addition, a top-down camera provides a global view of the whole scene, while cameras mounted on the two robotic arms offer local views to capture fine-grained details from their respective perspectives.

Manufacturing tasks performance

In our case study demo, we take a representative unexpected situation in manufacturing — 'assembly line accidentally stops' — to illustrate the proposed workflow. First, as shown in Fig. 3.6(1), through visual parsing and reasoning on the global view, the model determines that the assembly line has halted because the designated storage area for gears is empty. At the same time, it detects that extra gears are available in a nearby region.

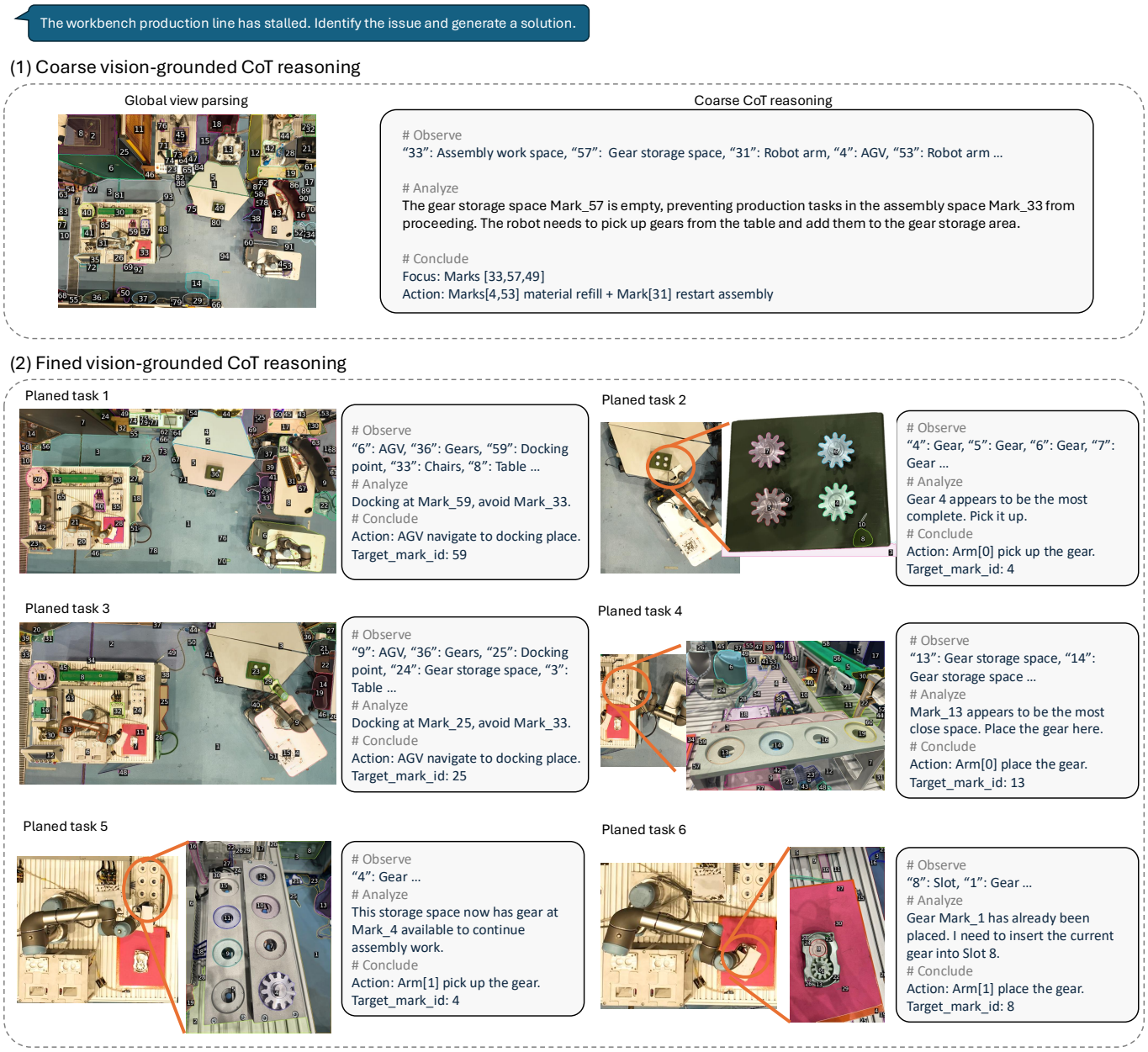


Fig. 3.6: Case study in real manufacturing environment.

Based on this observation, the system performs coarse CoT reasoning to identify the corresponding ROI. Next, the AGV and the two robotic arms proceed with step-by-step fine-grained reasoning to complete the tasks in sequence. Subsequently, as shown in Fig. 3.6(2), the AGV and the two robotic arms commenced sequential reasoning to accomplish the task step by step. According to the planned procedure, the mobile platform first replenished the materials. Following the detection of gear replenishment, the workstation-mounted arm resumed the assembly process from the point of interruption, thereby resolving the issue. The comprehensive coarse-to-fine vision-grounded reasoning approach effectively integrates global information with local details while leveraging the expansive knowledge of foundation models to address unforeseen problems. This paradigm holds significant implications for the advancement of human-centric smart manufacturing in the future.

Discussion

Deploying the proposed method in real-world settings requires substantial engineering and systems support, and these implementation details can have a nontrivial impact on overall performance. First, the accuracy of vision-grounded reasoning is sensitive to lighting conditions and camera resolution. Consequently, reliable industrial deployment depends on high-precision cameras, appropriate camera height and placement, with bright and stable ambient illumination. Second, when a single computer communicates with multiple robots, the system will encounter frequent disconnections, reduced throughput, and packet loss. Thus, practical deployment also demands a robust communication infrastructure. Finally, translating high-level task planning into well-specified, fine-grained subtasks—and coordinating each robot's manipulation and navigation execution—requires tight integration. Real industrial applications require extensive manual design and empirical tuning to achieve optimal system-level coordination.

3.4 Summary

To address the unforeseen challenges faced by smart manufacturing systems in open-world environments, we propose a Coarse-to-Fine Vision-Grounded Chain-of-Thought Reasoning Approach. By integrating multi-granularity visual parsing with a coarse-to-fine CoT framework, our method jointly leverages global scene context and local visual perspectives to enable interpretable, zero-shot reasoning, while generating clear and precise robot-executable task plans. Extensive experimental results demonstrate that our approach achieves outstanding performance in terms of success rate and generalization across complex and previously unseen scenarios.

However, our approach also has certain limitations. For instance, vision-grounded reasoning can be affected by factors such as lighting conditions and camera resolution, which may impact the accuracy of visual parsing. Additionally, due to the incorporation of intermediate analysis stages within the model, the overall reasoning process tends to be time-consuming, making it less suitable for scenarios that require rapid or real-time responses. Finally, our method does not yet incorporate closed-loop error correction or mitigation.

Our future investigations will prioritize improving the stability and dependability of vision-driven reasoning, incorporating richer production line information, and achieving a better balance between long-chain reasoning and rapid response capabilities. In addition, we plan to explore movable visual perspective capture to improve perception flexibility and adaptability in dynamic environments. Finally, VLM-based error detection and correction will be integrated to improve the system's robustness.

4 Vision-Language Model-based Human-guided Mobile Robot Navigation in an Unstructured Environment for Human-centric Smart Manufacturing

4.1 Introduction

Industry 5.0 represents a paradigm shift toward sustainable and resilient manufacturing systems, emphasizing HSM that places human well-being at its core [52]. Mobile robotics constitutes a fundamental component within smart manufacturing ecosystems, with widespread applications in inspection protocols, warehouse logistics, and material handling operations [69, 70]. While research on mobile robots in manufacturing has focused on localization, tracking, trajectory planning, and motion control, aiming to achieve fully automated and self-organized systems [71]. Current technological limitations impede the realization of completely self-organized manufacturing environments, particularly regarding system resilience and adaptability. These constraints become especially pronounced in unstructured environments characterized by unpredictable tasks, variable structural configurations, and irregular dimensional parameters, where conventional pre-programmed approaches prove inadequate [72].

Within complex operational environments, intelligent systems must deploy sophisticated methodologies for information acquisition, processing, and environmental cognition to establish reliable situational awareness [73, 74]. While structured automation lines represent the manufacturing ideal, their implementation remains impractical in numerous scenarios. The assembly of large-scale aircraft exemplifies this challenge, where the integration of components within and around massive, irregularly shaped fuselages currently necessitates manual intervention. The spatial constraints, coupled with task complexity and variability, preclude workers from maintaining immediate access to all requisite materials, necessitating auxiliary personnel for tool and component logistics, thereby compromising operational efficiency. Similarly, in maintenance operations, despite initial equipment preparation, the dynamic nature of inspection processes frequently reveals unforeseen requirements for additional tools or components. This iterative need for resource retrieval, often extending across multiple maintenance sessions, significantly impacts workflow continuity and operational effectiveness.

In alignment with Industry 5.0 principles, the integration of HRI into mobile robotic systems presents a viable solution to contemporary manufacturing challenges [75, 76]. This symbiotic approach harmonizes human cognitive strengths—including creativity and adaptive problem-solving capabilities—with robotic advantages in speed and endurance, facilitating the development of flexible and resilient HSM systems [16, 77, 78, 79]. The recent advancement of LLMs and VLMs has revolutionized vision and language-based HRI, offering unprecedented capabilities [80, 81, 79, 82]. These technologies demonstrate remarkable generalization to novel scenarios, enhance robotic perception through human-like multimodal processing, and establish intuitive communication interfaces [9]. Vision and language-based human-guided navigation emerges as a particularly promising paradigm for unstructured manufacturing environments characterized by dynamic, unpredictable conditions. This approach enables mobile robots to synthesize

natural language commands with visual environmental data, facilitating adaptive interaction with both human operators and physical surroundings. Such capabilities prove invaluable in complex assembly and maintenance scenarios, where robots can autonomously respond to engineers' evolving requirements for tool and component retrieval. This automation of ancillary tasks enables technical personnel to focus on specialized functions, substantially improving operational efficiency and resource utilization within manufacturing processes.

While existing mobile robot systems primarily rely on pre-programmed tasks and autonomous navigation algorithms in structured environments, they lack the flexibility to adapt to unfixed manufacturing scenarios and real-time human needs [70, 83, 84, 85]. In contrast to traditional approaches, VLM-based methods enable natural human-robot interaction and environmental recognition, allowing robots to understand and respond to spontaneous human commands in unstructured manufacturing environments. This integration of VLM with mobile robot navigation demonstrates superior adaptability in addressing unpredictable demands that may arise during manufacturing processes, particularly in complex scenarios such as large aircraft assemblies where traditional automation methods prove inadequate. However, previous research [86, 87, 39, 22] about vision and language-based HRI for robot navigation encounters predominantly two fundamental constraints. Firstly, most endeavors exhibit inadequate generalization capabilities regarding unseen objects and environments, as well as the diverse and variable natural language expressions of humans. This deficiency severely restricts their application in complex and variable manufacturing environments. Secondly, although most works perform well within simulators, in the real world, sensor noise can lead to a cluttered and confusing reconstruction of top-down maps, aliasing geometric details and impeding accurate segmentation of navigation targets. This limitation prevents their effective empowerment in the industry.

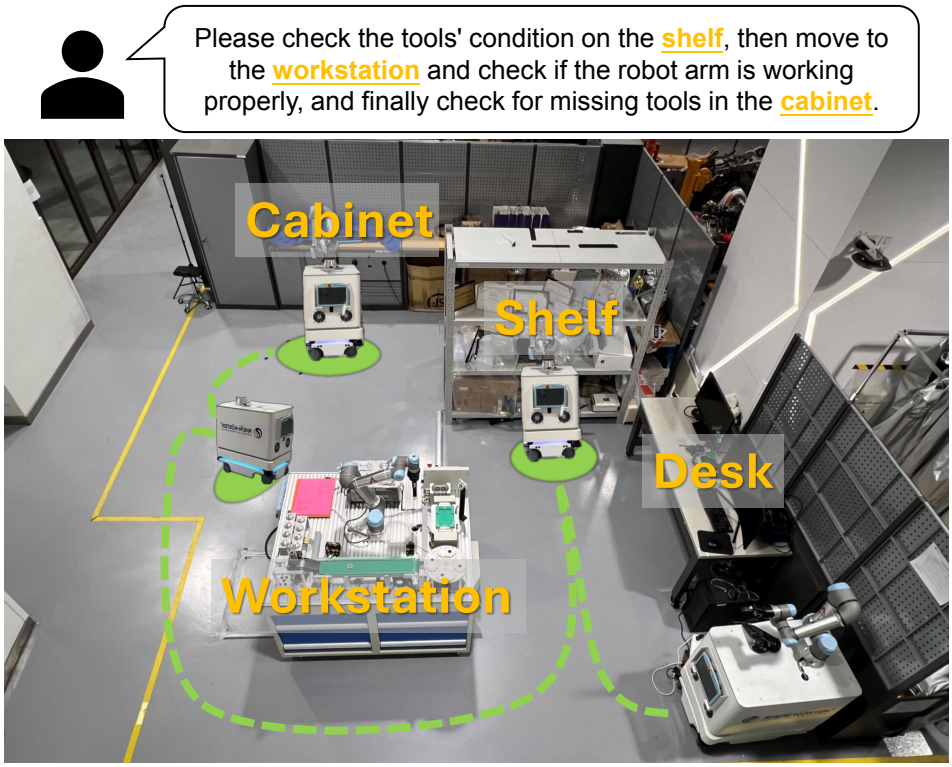


Fig. 4.1: Application example of proposed VLM-based human-guided mobile robot navigation approach in manufacturing.

To overcome the challenges, this research proposes a VLM-based human-guided mobile robot navigation approach in an unstructured environment for HSM. This system consists of three modules: robust three-dimensional (3D) scene reconstruction, VLM-based semantic segmentation and integration, and LLM-based spatial navigation. This work aims to facilitate human control over robot navigation for completing manufacturing tasks in unstructured settings through a natural interaction mode. Fig. 4.1 illustrates an example of our proposed method applied to an inspection task.

4.2 Methodology

To realize vision-and-language-based robot navigation in unstructured environments pertinent to HSM, we propose a VLM-based human-guided mobile robot navigation system designed to mitigate the impact of sensor noise on reconstructed maps under realistic conditions, as shown in Fig. 4.2. The input

of the system is RGB-D video frames as well as natural language instructions, while the output of the system is the Python code to control the robot to perform navigation tasks. The system consists of three main parts:

1. **3D robust scene reconstruction.** The first component of the system involves 3D robust scene reconstruction, where a pipeline including several popular point cloud registration methods is employed to reconstruct the scene point cloud based on input RGB-D videos and mitigate the impact of noise on geometric structure reconstruction. Based on the reconstructed scene's mesh model and the robot's altitude, an obstacle map is calculated for subsequent robot navigation path planning.

2. **Semantic segmentation and integration.** The second part of the system involves semantic segmentation and integration. A VLM is utilized for language-driven semantic segmentation of RGB images in the input video. Subsequently, the semantic information is integrated into the reconstructed 3D scene mesh using the RGB-D integration method to obtain 3D scene semantics.

3. **Spatial goal navigation.** The third component of the system involves spatial goal navigation. An LLM is employed to comprehend natural language commands from humans. By designing the prompts and integrating 3D scene semantics, natural language is translated into robot control code to trigger the robot to perform navigation actions.

4.2.1 3D robust scene reconstruction

One key element in reconstructing a 3D scene is the accurate estimation of camera pose, which directly impacts the precision of the reconstruction results. In manufacturing environments, where the surroundings are often cluttered and noisy, precise estimation of the camera pose on a mobile robot

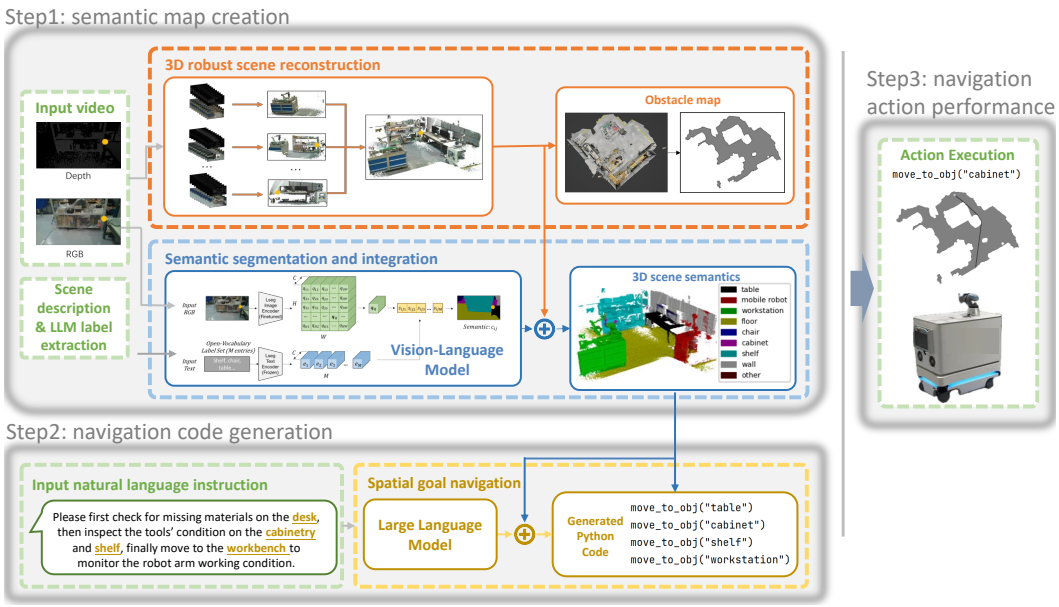


Fig. 4.2: Structure of proposed VLM-based human-guided mobile robot navigation system.

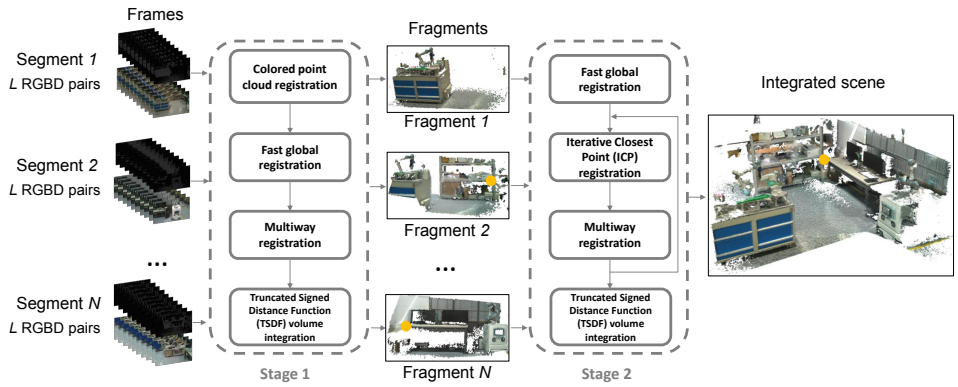


Fig. 4.3: Reconstruction pipeline [88].

using sensors such as Inertial measurement units (IMUs) and accelerometers becomes exceedingly challenging and fails to meet the required accuracy. Therefore, this research utilized visual odometry calculation based on RGB-D image information to estimate camera pose. Specifically, this research employs a pipeline [88] comprising multiple point cloud registration methods to mitigate noise and visual odometry drift for precise and robust scene reconstruction, as shown in Fig. 4.3.

In Fig. 4.3, the entire reconstruction process is divided into two stages to reduce the impact of odometry drift. In the first stage, the input video frames

are divided into N segments, each containing L pairs of RGB-D images. The reconstruction process is performed on each segment, resulting in N point cloud fragments. The reconstruction process in the first stage comprises four essential steps: colored point cloud registration [89], fast global registration [90], multiway registration [88], and TSDF (Truncated Signed Distance Function) volume integration [91]. The colored point cloud registration finds the camera movement between two consecutive RGB-D image pairs based on color and geometry information. Fast global registration aligns non-consecutive pairs based on geometry information. Multiway registration could prune erroneous constraints created by global registration for robust optimization. And finally, all the frames in this sequence are fused to a fragment mesh model by TSDF volume integration. The second stage entails applying a similar reconstruction process on these N fragments to generate a complete 3D scene mesh, including fast global registration [90], ICP registration (Iterative Closest Points) [92], multiway registration [88], and TSDF volume integration [91]. The fast global registration roughly aligns the fragments as the initialization before ICP registration. Then point-to-plane ICP registration is applied to do the tight alignment, followed by multiway registration to prune erroneous constraints. The ICP registration and multiway registration are applied twice to refine the fragments' alignment. Finally, based on the optimized pose graph, TSDP volume integration is applied to fuse all the video frames to a whole mesh model.

The obstacle map is a binary map derived from the reconstructed 3D scene and calculations of the robot's dimensions, serving as the foundation for obstacle avoidance and path planning for subsequent robot navigation. Adapting the obstacle map to the specific size of the robot is of vital importance. For example, a 1.5-meter-wide passageway is considered an available area for a robot with a 1-meter diameter but considered an obstacle for a robot with a 2-meter diameter. The obstacle map calculation method is very mature and

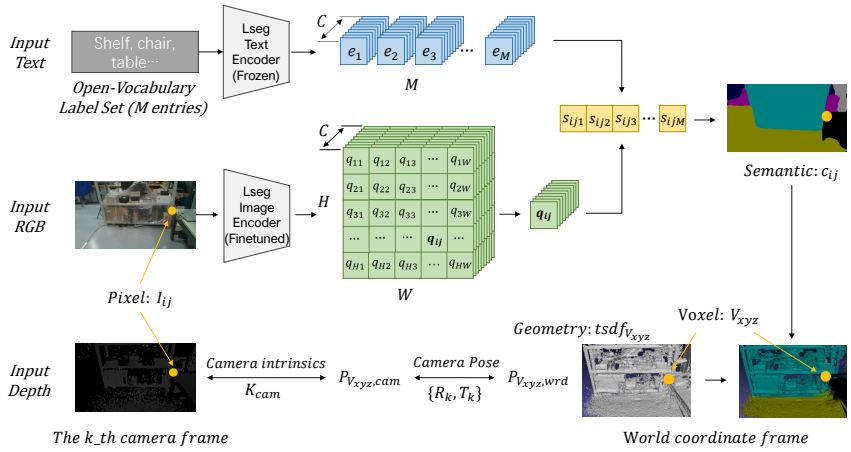


Fig. 4.4: Semantic segmentation and integration.

can be easily obtained from the simulator, such as the AI Habitat simulator [93], and one may refer to [94, 95] for more details.

4.2.2 Semantic segmentation and integration

Based on the aforementioned work, one core contribution of this research lies in the fusion of VLM with optimized 3D reconstruction results to achieve a zero-shot understanding of the 3D scene. To achieve this objective, this work employs RGB data to represent the semantic information derived from the VLM segmentation of each RGB image, and integrate the semantic information into the point cloud surface reconstruction phase using the TSDF algorithm, as shown in Fig. 4.4.

Semantic segmentation

The semantic segmentation part in our work utilizes the LSeg model [96] to perform language-driven semantic segmentation on all RGB frames within the video. LSeg is a VLM, in which the text encoder is the CLIP text encoder, it is utilized to generate the text embeddings of the input text labels, such as $workstation$ and $floor$, and allows for the free-form description of these input labels. In this work, text labels are extracted by an LLM from human natural language descriptions of the environment, pinpointing navigation

targets. For instance, when humans describe the environment as *'a shopfloor with a cabinet, a table, a shelf, and a workstation'*, the LLM identifies *'cabinet, table, shelf, workstation'*, facilitating human-centric interaction in smart manufacturing. The image encoder in LSeg is transformer-based, it can generate pixel-level image embeddings. By measuring the closeness of text feature vectors and image feature vectors, the semantic content can be derived.

In this work, the fine-tuned LSeg's visual encoder $f(I) : R^{H\times W\times 3} :\rightarrow R^{H\times W\times C}$ is applied to RGB image frame I^k to get pixel level visual embedding $F_k \in R^{H\times W\times C}$. For each pixel I_{ij}, the corresponding visual embedding is

$$q_{ij} = F_k(i, j), \tag{4.1}$$

where $q_{ij} \in R^{1\times C}$. Similarly, the pre-trained CLIP text encoder is applied to M text labels to obtain the text embeddings

$$E = [e_0, e_1, ..., e_M], \tag{4.2}$$

where $E \in R^{M\times C}$. Then the inner product is performed on image and text embeddings to compute the pixel-to-category similarity matrix

$$S_{ij} = q_{ij} \cdot E^T, \tag{4.3}$$

where $S_{ij} \in R^{1\times M}$. Within the similarity matrix, the position holding the largest value denotes the pixel's category, revealing its semantic meaning. Different RGB values are utilized to represent distinct semantic information. Therefore, the semantic information of the pixel I_{ij} is c_{ij}, which is an 8-bit RGB value.

Semantic integration

The conversion between camera and world coordinate systems is fundamental for relating two-dimensional (2D) image pixels to 3D voxel structures. The mapping between the camera coordinate frame and the world coordinate system is

$$P_{wrd} = R_k P_{cam} + T_k, \tag{4.4}$$

where R_k, T_k means the precise camera pose which has been calculated and optimized in the 3D point cloud robust reconstruction, P_{cam} is the camera coordinate, and P_{wrd} is the world coordinate. By employing this transformation, the semantic information of each frame can be integrated with the 3D geometric information.

After that, TSDF integration is applied to integrate all the frames into the global point cloud. For a voxel V_{xyz} located in (x, y, z) in the global point cloud, its coordinate in world coordinate is $P_{V_{xyz},wrd}$ which can be obtained according to voxel length. Based on equation (4.4), the corresponding camera coordinate $P_{V_{xyz},cam}$ can be calculated. Hence, the depth of the voxel V_{xyz} relative to the camera $Cam_z(V_{xyz})$ can be easily obtained. The principle of camera imaging is

$$Cam_z(V_{xyz}) \cdot I_{ij} = K_{cam} \cdot P_{V_{xyz},cam}, \tag{4.5}$$

where I_{ij} means the pixel, K_{cam} means the the camera intrinsics matrix. Based on that, the corresponding pixel I_{ij} is found. Thus, the sdf of V_{xyz} is

$$sdf(V_{xyz}) = D(I_{ij}) - Cam_z(V_{xyz}), \tag{4.6}$$

where $D(I_{ij})$ is the depth value of pixel I_{ij}. Therefore, tsdf value of this voxel $tsdf_{V_{xyz}}$ is

$$tsdf_{V_{xyz}} = max[-1, min(1, sdf(V_{xyz})/t)], \tag{4.7}$$

where t is tsdf_trune = 0.04. Therefore, the tsdf value $tsdf_{V_{xyz}}$ and semantic information c_{ij} are stored in a voxel V_{xyz}. By averaging the tsdf value $tsdf_{V_{xyz}}$ and semantic value c_{ij} of each frame within this voxel V_{xyz}, the overall information of the entire model can be obtained:

$$tsdf_{V_{xyz}} = \frac{1}{N} \cdot \sum_{k=1}^{N} tsdf_{V_{xyz}}(k) \tag{4.8}$$

$$c_{ij} = \frac{1}{N} \cdot \sum_{k=1}^{N} c_{ij}(k). \tag{4.9}$$

The areas where the TSDF value is zero represent the surface of the object. This can be identified using the Marching Cubes algorithm. By assigning colors based on the semantic information values, the final 3D semantic map is generated. In the generated 3D scene semantic mesh, all vertices belonging to the same object have the same RGB value. By averaging the coordinates of these vertices with the same RGB value, the center coordinates of the object can be obtained. In this way, 3D scene semantics are generated, which include the semantic description for each navigation object and its corresponding center coordinate.

4.2.3 Spatial goal navigation

In spatial goal navigation, the GPT-3.5 model is utilized to convert instructions into executable Python code, as shown in Fig. 4.5. To achieve this, a set of initial prompts is established to outline the VLN task. Subsequently, dialog examples are provided to illustrate the mapping between inputs and outputs. User instructions are then incorporated. The prompt is sent to the GPT model via an API to generate the Python code. The Python function $move\_to\_obj(object)$ is predefined, including object coordinate indexing, path planning, and controlling the robot to move to the object from its current

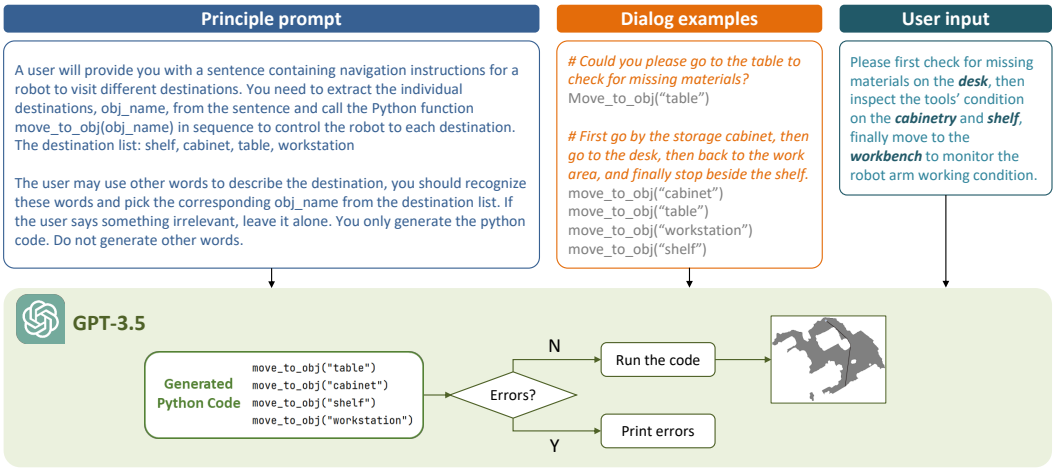


Fig. 4.5: LLM prompt.

position. LLM can identify each navigation object from the natural language instruction and generate the code to call the Python function. If there are no compilation errors, the code is executed to carry out the robot navigation actions.

4.3 Experiments and discussions

This section presents three independent experiments, each addressing a specific system component, to assess the method's feasibility and effectiveness.

4.3.1 Evaluation of semantic segmentation

LSeg is selected as the model for zero-shot semantic segmentation to perform semantic segmentation on the scene's 2D RGB images. The semantic segmentation experiment is divided into two parts: the first part focuses on the conventional semantic segmentation capability assessment, while the second part focuses on the zero-shot capability evaluation.

Experiment settings

The constructed segmentation dataset contains 175 samples, of which 105 are designated for training, 35 for validation, and 35 for testing. Notably, the scenes represented in the test set were distinct from those encountered during the training phase. The LSeg image encoder is fine-tuned on the custom dataset using a Nvidia RTX 3090 GPU to ensure it can accurately segment targets within industrial scenes; meanwhile, the LSeg text encoder remains frozen to preserve the model's zero-shot capabilities.

Experiment results

To validate the accuracy of semantic segmentation, this work conducts comparative experiments between LSeg and two fully supervised semantic segmentation models, U-Net [97] and DeepLabV3+ [98], all of which are fine-tuned using the same dataset. U-Net [97] has a symmetric U-shaped structure and skip connections of feature maps, which enable the model to effectively combine contextual information and detailed features of images during segmentation tasks. DeepLabv3+ [98] is known for its robust performance in various segmentation challenges, which employs atrous convolution and an encoder-decoder structure to enhance segmentation accuracy. This research evaluates the models using pixel accuracy. The semantic segmentation comparative experiment result is illustrated in Table 4.1. UNet and DeepLabv3+, while designed for fully supervised learning, show expectedly strong performance due to their direct training on annotated images. Remarkably, LSeg demonstrates comparable results to the fully supervised models, which is an impressive feat considering its zero-shot learning capability.

Table 4.1: Semantic segmentation performance on custom dataset

Method	Backbone	Pixcel Accuracy	
		Val	Test
U-Net [97]	VGG	92.82%	83.34%
DeepLabv3+ [98]	MobileNet	94.37%	80.46%
LSeg [96]	ViT-L/16	96.16%	87.55%

LSeg demonstrates zero-shot performance by producing semantic segmentations guided by textual prompts absent from its training data, primarily enabled through the integration of the CLIP text encoder. The CLIP model maps visual and textual information into a shared embedding space, bringing semantically similar items—such as labels and their corresponding visual representations—closer together. This allows LSeg to directly use text embeddings for segmentation without specialized training for each possible category. Benefiting from diverse training corpora and robust learning capacity, the text encoder in CLIP organizes the embedding space into a coherent semantic structure. This means that even for labels not seen during training, as long as they are semantically related to labels present in the training data, the model can find similar representations in the embedding space, thereby achieving generalization to these new labels[96]. The labels employed to refine the image encoder encompass the following items: *table, chair, workstation, shelf, automated guided vehicle, cabinet, floor, wall, background*. Notably, when diverse terms are utilized to denote the same objects, accurate recognition is achieved, as illustrated in Fig. 4.6.

In Fig. 4.6, (a) displays the results of semantic segmentation indexed by the original training labels, while (b), (c), and (d) illustrate the results of semantic segmentation indexed using labels not encountered during the training process. In Fig. 4.6 (b), the labels *shelving*, *robot*, and *floor* are unseen labels, yet the model can accurately segment them. Notably, the segmentation result for the *robot* label is remarkable. In Fig. 4.6 (a), within the original training dataset, the labels *workstation* and *automated guided vehicle* are employed, both of which encompass elements of a *robot arm*, thereby falling under the category of *robot*. In Fig. 4.6 (b), the *robot* label is solely used, enabling the model to successfully segment both objects as *robot*, demonstrating its robust zero-shot capabilities. In 4.6 (c), new labels such as *desk*, *floor*, *mobile robot*, and *HRC workstation* have been employed for textual descriptions in semantic segmentation, and the model

has successfully and accurately segmented these targets as depicted in the figure. Noteworthy are the labels *HRC workstation* and *mobile robot*, which significantly differ from the original training labels *workstation* and *automated guided vehicle*, yet the model has successfully correlated and segmented them accordingly. Particularly, in 4.6 (b), the segmentation of *mobile robot* does not categorize all objects containing *robot* elements but successfully interprets the meaning of *mobile*, thereby isolating only the *automated guided vehicle*. This demonstrates the model's formidable comprehension capabilities. In 4.6 (d), the unseen labels include *desk*, *floor*, *AGV*, and *HRC workstation*, and the model has correctly completed the segmentation. Notably, the label *AGV*, which stands for *automated guided vehicle*, consists of an abbreviation rather than a full word to convey semantic information. Nevertheless, the model has successfully associated the abbreviation *AGV* with its full term, demonstrating the model's powerful inferential capabilities.

The demonstrated robust generalization ability of LSeg permits humans to refer to objects in an unstructured manufacturing scene according to their own conventions, eliminating the need to predefine and adhere strictly to specific names for each object. This advancement brings the entire system closer to a more natural HRI.

4.3.2 Evaluation of 3D reconstruction

To mitigate the impact of sensor errors on the reconstruction outcomes, a pipeline comprising several point cloud registration and integration methods is employed in this system. This experiment aims to validate the accuracy of 3D reconstruction under this pipeline, assessing it from both quantitative and qualitative perspectives.

Experiment settings

Data are collected by an Intel RealSense L515 camera in real-world settings including three areas, each comprising hundreds of frames of RGB-D video for scene reconstruction. During the reconstruction process, a Nvidia RTX 3080 is employed to accelerate the computation.

Throughout the reconstruction process, the RGB-D sequence is divided into multiple segments, each containing 100 frames. Every fifth frame is selected as a keyframe for odometry estimation and optimization, balancing computational efficiency with registration accuracy. Valid depth values between 0.3 m and 3 m are considered to filter out invalid data that are too close or too distant from the camera, thereby reducing noise and errors. A voxel size of 0.05 m is employed for point cloud downsampling and TSDF fusion. During registration, the maximum acceptable distance threshold between corresponding point pairs is set to 0.07 m to reduce mismatches by filtering out pairs with excessive discrepancies. The fitness threshold for registration results is set at 0.3. Only when the fitness exceeds this value is the registration considered valid, ensuring that only high-quality registrations are accepted while unreliable ones are rejected. In SLAC optimization, the maximum number of iterations is limited to five to ensure completion within reasonable computation time and to prevent overfitting and excessive computation. For TSDF fusion, the cubic size is set to 3 m to define the dimensions of the reconstruction space. Additionally, the truncation distance in TSDF fusion is specified as 0.04 m. This truncation threshold affects the level of detail in surface reconstruction and the overall fusion effectiveness.

Experiment results

For quantitative evaluation, iPhone-based LiDAR scanning results are employed as the ground truth, with the Chamfer score between the reconstructed point sets and the ground truth calculated to verify the accuracy of the scene reconstruction. The Chamfer distance is a method for measuring the distance between two sets of points, representing the aggregate of distances from one

set to another. It is commonly used to compare the similarity between two geometric shapes. The formula for calculating the Chamfer distance $d_{CD}(A, B)$ between point cloud $A = \{a_1, a_2, ..., a_m\}$ and point cloud $B = \{b_1, b_2, ..., b_n\}$ is

$$d_{CD}(A, B) = \frac{1}{|A|} \sum_{a \in A} \min_{b \in B} \|a - b\| + \frac{1}{|B|} \sum_{b \in B} \min_{a \in A} \|b - a\| \tag{4.10}$$

The smaller the Chamfer distance, the greater the similarity between the two point clouds. This work evaluates the reconstruction effectiveness of the system by calculating the Chamfer distance between the point cloud reconstructed from RGB-D frames and the ground truth. The reconstruction visualization and experiment results are provided in Fig. 4.7 and Table. 4.2.

Table 4.2: 3D reconstruction performance on custom dataset

Scene	Number of frames	Chamfer distance (meter)	Reconstruction time (minute)
Area 1	627	0.071	9.20
Area 2	804	0.081	11.08
Area 3	408	0.054	8.03
Average	613	0.069	9.38

The entire scenes are scanned using LiDAR of an iPhone and the reconstructed point clouds are illustrated in Fig. 4.7 (a), (d). Subsequently, the scenes are divided into three distinct areas, and the reconstruction is performed based on RGB-D for each area, the point clouds of the three reconstructed areas are depicted in Fig. 4.7(b), (c), and (e). Table 4.2 presents the Chamfer distance between the reconstruction outcomes of the three areas and their corresponding ground truth, utilized to estimate the accuracy of the 3D reconstructions based on RGB-D frames. The Chamfer distance for each area does not exceed 10 cm, with the average Chamfer distance being 6.9 cm. Given that the two entire scenes encompass more than 10 square meters, the

reconstruction error is comparatively within an acceptable range. In addition, the reconstruction times for each region are detailed in Table 4.2, averaging approximately 9.38 minutes. Consequently, an acceptable trade-off between reconstruction speed and accuracy has been achieved.

For the qualitative comparative experiment, the baseline model is VLMaps [31] as they use a similar framework to ours, but they do not account for sensor noise in real-world industrial scenarios in map reconstruction. Since VLMaps does not execute a complete 3D reconstruction but instead directly produces a reconstructed 2D top-down map, the comparison of reconstruction outcomes is confined to qualitative assessment through images obtained from identical datasets. The qualitative comparative experiment results are provided in Fig. 4.8.

In the comparative experiment, 3D reconstructions are conducted based on the same dataset by both methods, and then the same fine-tuned LSeg model is applied for semantic segmentation of RGB images within the dataset, finally, the segmentation results are integrated into the reconstructed point clouds. In Fig. 4.8, the left column displays images of 3D reconstructions incorporating geometric information, while the right column presents images of 3D reconstructions enriched with semantic information. As illustrated in Fig. 4.8(a), the geometric details in the VLMaps reconstruction results are considerably disordered due to the absence of any camera pose optimization technique, affected by errors in camera pose estimation. Even though LSeg successfully segments navigation targets from RGB images, the reconstructed map is unsuitable for real-world applications. In contrast, Fig. 4.8(b) shows our reconstruction results, where the geometric details are remarkably clear, and the semantic information is accurately integrated, significantly empowering tasks related to VLN.

4.3.3 Spatical goal navigation

Experiment settings

The spatial goal navigation experiments are conducted within the AI Habitat simulator [93], which is an open-source simulation platform that offers a highly realistic 3D environment. This platform enables AI agents to test a variety of tasks, including navigation and human-robot interaction. In this experiment, GPT-3.5 is linked with the simulator, wherein humans input natural language commands through a dialogue box. GPT-3.5 then generates and executes the code to control the robot's movement based on the given instructions. Subsequently, the robot performs path planning and navigates to the target location. Upon completing the entire set of instructions, the robot returns a video showcasing the entire movement process from a first-person perspective. The primary focus of this work is on enabling the robot to comprehend language instructions and autonomously identify all navigation target points. Therefore, the process of the robot moving to the subgoals is not concentrated. This work assumes that once the coordinates of the subgoals are identified, the robot will successfully move to those locations. Consequently, the path-planning algorithm is not the emphasis of this work, and this research directly employs the Pathfinding module provided by the simulator to accomplish the path planning.

Experiment results

This experiment tests whether GPT-3.5 could accurately generate the corresponding code and successfully execute it using different natural language instructions. Specifically, the natural language instructions are categorized based on the number of navigation subgoals contained within them, resulting in four categories, encompassing the number of subgoals from 1 to 4. For each category, the tests are conducted with 20 distinct natural language

instructions and tallied the success rates. The results are displayed in the Table 4.3.

Table 4.3: Spatial goal navigation performance

No. Subgoals	Success rate
1	20/20
2	19/20
3	19/20
4	16/20
Average	92.5%

The data in Table 4.3 represents the overall navigation success rate of the system. As natural language instructions include more subgoals, the overall success rate declines, averaging 92.5%. During the experiments, it is discovered that errors primarily occur when the sequence of subgoals mentioned in the natural language instructions does not align with the actual intended sequence. For instance, if the input command is *'come to the shelf, before that go to check the equipment on the table'*, the LLM would direct the robot to first approach the shelf and then the table, which is contrary to the human's desire for the robot to visit the table before the shelf. However, changing the sequence does not necessarily lead to errors. For example, if the input command emphasizes the order more clearly with *'come to the shelf, but before that, firstly you should go to check the equipment on the table'*, the LLM would correctly understand the sequence, and direct the robot to visit the table before the shelf.

4.3.4 Case study

Finally, a case study in an industrial setting is presented to validate the proposed VLM-based human-guided mobile robot navigation system in HSM. The experimental scenario, depicted in Fig. 4.1, takes place in a smart manufacturing laboratory consisting of a workstation, cabinet, shelf, and desk. The workstation serves as a human-robot collaborative assembly bench,

while the cabinet, shelf, and desk store various parts. Fig. **??** illustrates the process of the pick-and-place case study.

In Fig. 4.9, a human gives a natural language command, "Take the tape from the cabinet to the shelf, then take the gear from the table to the workstation." Upon identifying the navigation targets, the LLM generates navigation code and executes it line by line, guiding the mobile robot to the destinations to complete the pick-and-place tasks. For each line of code, the coordinates of the navigation targets are derived from the 3D scene semantics. Traditional localization and path planning are handled by invoking the existing AGV APIs. It is important to note that this task focuses solely on navigation, with the AGV navigation code generated by the LLM based on human verbal commands, while the code related to the robotic arm is pre-configured to provide a comprehensive demonstration of the case study.

This pick-and-place case study validates the effectiveness of the proposed VLM-based human-guided mobile robot navigation system in unstructured HRI and smart manufacturing scenarios. The case study demonstrates the system's significant potential in real industrial applications, showcasing its ability to effectively enhance the efficiency of HRI and HRC, thereby enabling the realization of human-centric smart manufacturing for the future of Industry 5.0.

4.4 Summary

To improve the resilience and flexibility of mobile robot navigation systems, and further realize HSM in Industry 5.0, this article presented a VLM-based human-guided mobile robot navigation approach, including 3D robust scene reconstruction, VLM-based semantic segmentation and integration, and LLM-based spatial goal navigation. The main contributions of this work lie in two aspects: 1) A VLM and LLM integrated approach was proposed to enable

humans to guide mobile robot navigation in unstructured manufacturing environments using free-form natural language; 2) an enhanced VLN framework was explored by integrating a pipeline featuring a series of point cloud registration and fusion methods, which significantly mitigates the impact of sensor noise on 3D reconstruction.

The main limitation of this work lies in the sole focus on the navigation of the mobile robot without considering the operational phase upon reaching the navigation destination. Additionally, there are other limitations, such as reliance on action primitives and the lack of updates for dynamic scenes. In the future, to further empower smart manufacturing with VLMs and LLMs, several directions are highlighted: 1) LLMs can be applied directly to the trajectory planning of robotic arms and end-effectors, enabling robot systems based on LLMs to perform more granular manipulation tasks; 2) Future initiatives could explore the incorporation of real-time environmental updates to further optimize VLN systems; 3) Integrating Augmented Reality (AR) into VLN interactions may largely enhance the system's interactivity in the future; 4) Focus on enhancing system security, stability, and other related factors to develop more reliable human-centric applications.

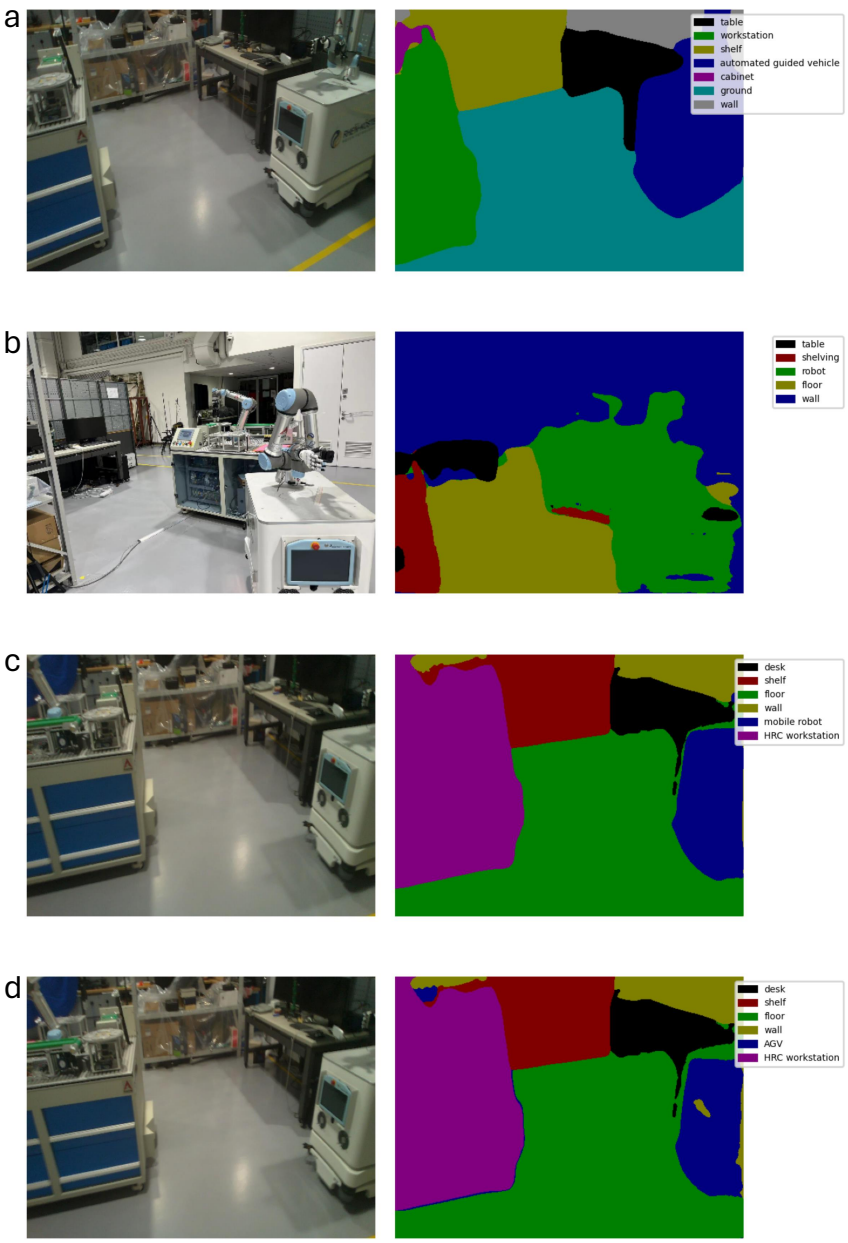


Fig. 4.6: Zero shot ability.

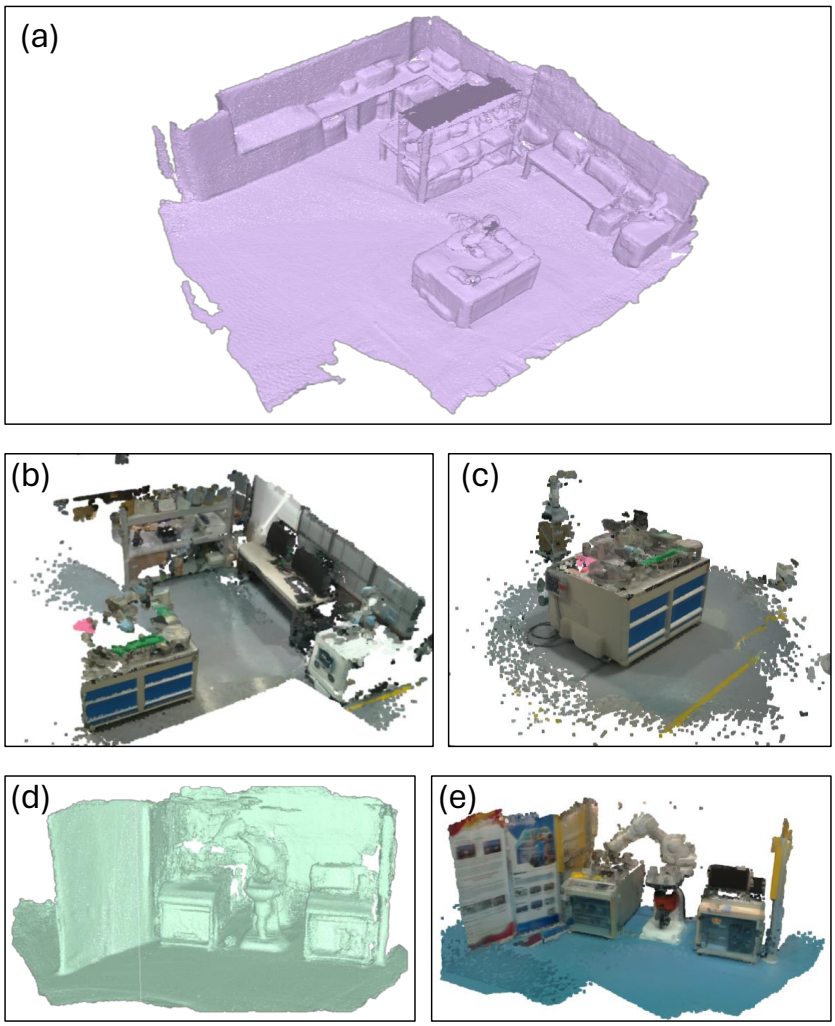


Fig. 4.7: Reconstruction visualization.

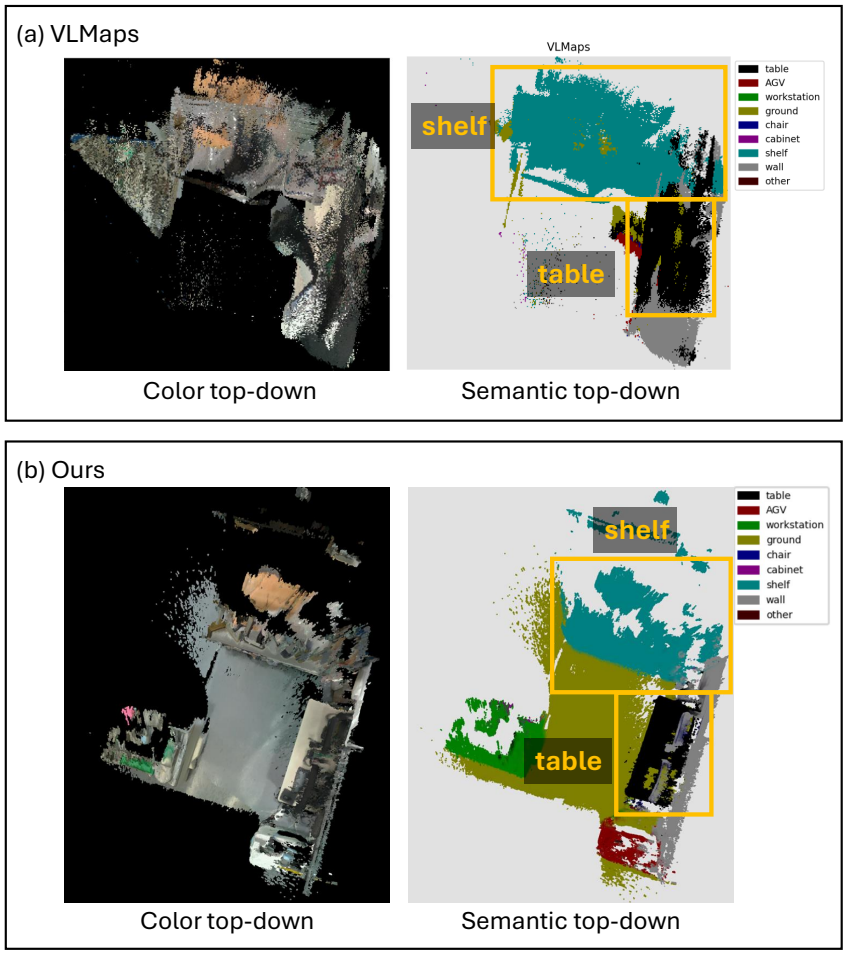


Fig. 4.8: Comparison of the 3D reconstruction results based on the same dataset between our system and the baseline method VLMaps[31].

.

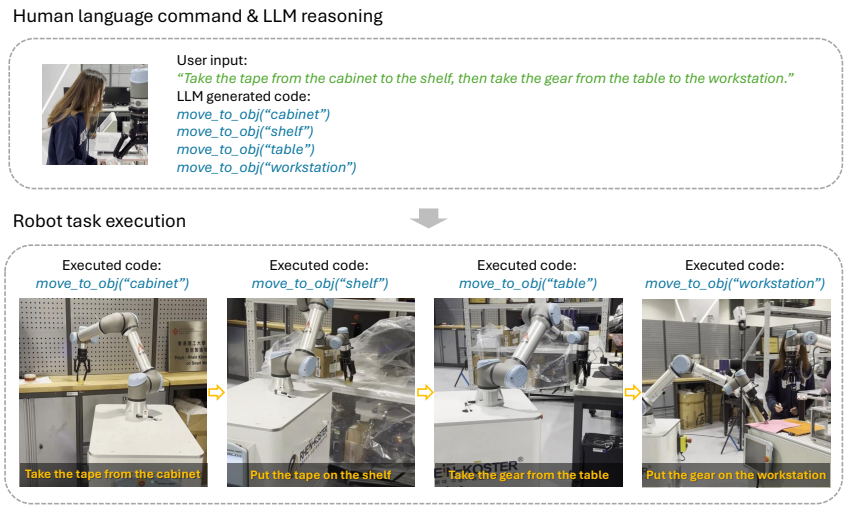


Fig. 4.9: Case study: vision and language-based HRI for pick and place work in smart manufacturing scenario.

.

5

3D-HiAR: A Foundation Model-based 3D Hierarchical Affordance Reasoning Approach for Industrial Robot Tool-use Manipulation

5.1 Introduction

Industry 5.0 prioritizes human well-being in manufacturing systems, emphasizing sustainability, resilience, and Human-centricity in smart manufacturing [52, 99, 100].In this context, HRI and HRC become indispensable, where robots and humans leverage their respective strengths through effective communication to accomplish tasks in close proximity [16, 39, 20]. Thus, as a widely practiced activity in the manufacturing field, it is crucial to investigate how robots can perform manipulation based on human verbal commands [61, 101, 102]. Current research in manipulation largely emphasizes relatively simple tasks based on the 'pick-and-place' action primitive, such as sorting, basic assembly, and material handling [103]. However, within manufacturing processes, numerous complex tasks cannot be accomplished merely by combining basic 'pick-and-place' actions; robot tool use is one of these critical tasks, particularly in the aircraft assembly process [104, 105, 106]. For instance, the assembly of fuselage and wing skins extensively utilizes riveting techniques. This involves using rivet guns to hammer and drive rivets into predrilled holes, ensuring secure structural connections. Additionally, avionics

installation and the fitting of seating and cabin door hinges involve numerous screw-driving tasks. The engine and landing gear sections of aircraft contain numerous bolted connections, whose installation processes rely on suitable wrenches and laser alignment tools for precision bolt fastening. Presently, these tool-dependent tasks are primarily performed manually. However, the confined spaces within aircraft structures (such as the interior of wings) and high-load scenarios (such as engine bolts) present extreme challenges for manual operations. Manual riveting and bolt tightening are prone to causing muscle strain, heightening occupational health risks. Furthermore, manual operations are susceptible to fatigue and variability in expertise, potentially resulting in uneven rivet formation, torque deviations, or insufficient bolt pre-tension, which pose serious safety hazards. Thus, exploring robotic tool-use tasks carries significant importance.

When humans aim to employ tools to accomplish a task, they naturally engage in these three stages of thought [107]. 1) How to hold the tool: Initially, it is essential to understand which part of the tool the hand should grasp. 2) How to contact the tool with the part: Subsequently, one must know which section of the tool makes contact with the component and the manner in which this contact should occur. 3) How to move the tool: Once the tool is in contact with the component, it is critical to comprehend how to maneuver the tool to complete the entire task. To enable robots to emulate these three stages of human thought, affordance reasoning is crucial. Originally described as recognizing the region of an item that permits interaction, this concept enables robots to infer how the object can be used and what actions it affords [108]. Foundation models endow affordance tasks with enhanced generalization capabilities, diminishing the dependence on limited affordance demonstration data [109, 80]. They also offer a natural HRI interface, enabling workers to directly control robots using natural language [110, 39, 82]. Thus, the three stages of human thought in using tools can be mapped to three affordances in robotic tool use: 1) Graspable Affordance:

Represents the contact points where the robot's end effector interacts with the tool. 2) Functional Affordance: In the current task, this refers to the contact points and contact posture between the tool and the component once the robot has grasped the tool. 3) Movement Affordance: After the tool is correctly in contact with the component, this involves the trajectory of the end effector's movement as the robot maneuvers the tool to complete the task.

Based on the above considerations, we propose 3D-HiAR, a foundational model-based 3D Hierarchical Affordance Reasoning approach for robotic tool-use manipulation in industrial tasks. 3D-HiAR mainly includes two parts: 1) 3D Joint Tool-Part Affordance (JTPA) reasoning, wherein a novel VLM with multi-dimensional feature fusion is designed for open-vocabulary 3D joint tool-part affordance detection, generating graspable affordances and functional affordances; and 2)Task-aware Trajectory Generation (TaTG), where an LLM is prompted to leverage geometry elements and movement affordances to generate end-effector trajectories according to different task instructions.

5.2 Methodology

In this article, 3D-HiAR is proposed, which is a foundation model-based 3D hierarchical affordance reasoning approach for robot tool-use manipulation tasks for industrial scenarios. The framework of 3D-HiAR is shown in Fig. 5.1. In this framework, the input consists of linguistic commands, such as "use the spanner to screw the bolt," combined with RGB-D observations. The output comprises the trajectory of the robotic arm's end-effector, detailing its position, orientation, and velocity. 3D-HiAR approach comprises two key components: 1) 3D Joint Tool-Part Affordance (JTPA) reasoning, where a novel VLM with multi-dimensional feature fusion is proposed to jointly infer graspable affordance and functional affordance based on task semantics

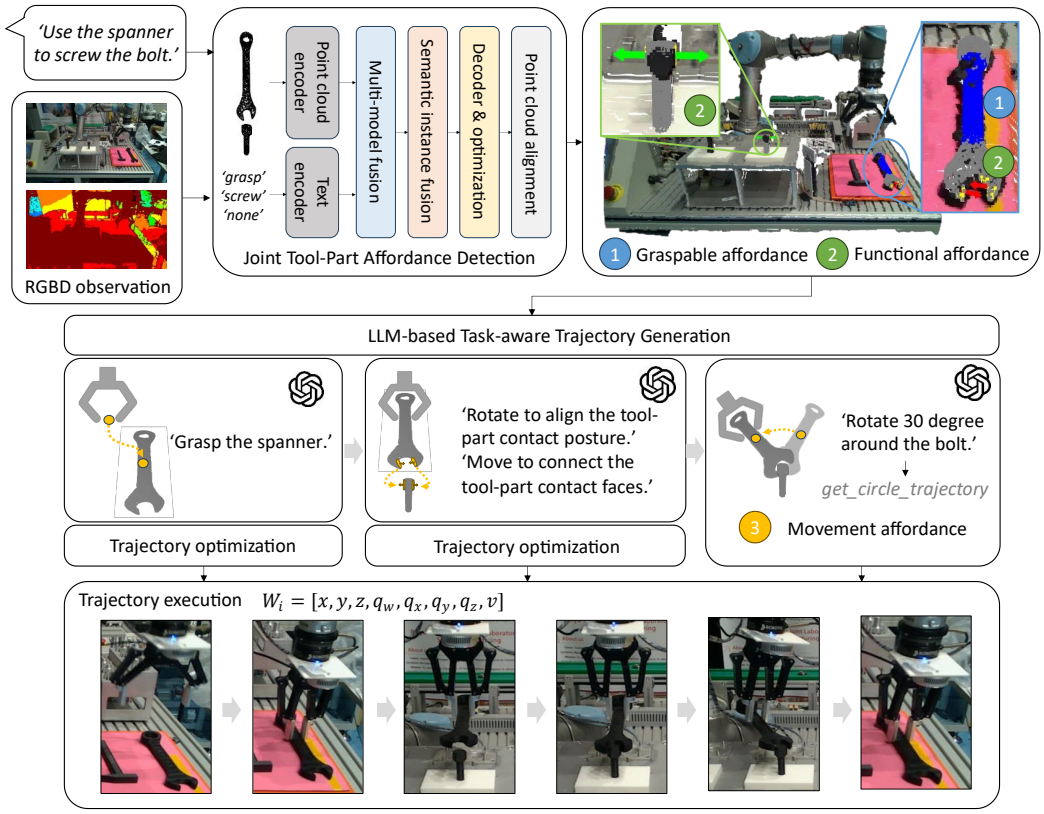


Fig. 5.1: The framework of 3D-HiAR: A Foundation Model-based 3D Hierarchical Affordance Reasoning Approach.

and 3D geometric relationships between tools and parts; and 2) Task-aware Trajectory Generation (TaTG), based on key geometric elements generated by graspable and functional affordances, and the movement affordance representing the trajectory shape, an LLM is employed to comprehend input task instructions and generate waypoints for the robotic arm's end effector trajectory.

5.2.1 Problem Formulation

In this tool-use task, the objective is to process RGB-D frames alongside language instructions to ultimately derive the trajectories of a robot manipulator's end-effector through the proposed 3D-HiAR framework. Initially, an LLM extracts m_l textual affordance labels $L = \{L_1, L_2, ..., L_{m_l}\}$ from the input language instruction Γ. Grounding DINO [111], in conjunction with

the Segment Anything Model (SAM) [112], performs zero-shot semantic segmentation on the RGB frame to generate 2D masks for both the tool and the part. These masks are then utilized with the depth frame to construct individual point clouds for the tool and the part. After 6D pose estimation and point cloud registration, the tool point cloud $C^t = \{c_1^t, c_2^t, ..., c_{n_c}^t\}$ and part point cloud $C^p = \{c_1^p, c_2^p, ..., c_{n_c}^p\}$ are constructed, $c_i^t, c_i^p \in \mathbb{R}^3, i = 1, ..., n_c$. The extracted textual affordance labels L and point clouds C^t, C^p then serve as the input for JTPA module and TaTG module, subsequently generating graspable, functional, and movement affordance representations A^g, A^f, A^m. These representations function as constraints and objectives within an optimization problem, ultimately facilitating the generation of the manipulation trajectory waypoints $\mathcal{W} = \{\boldsymbol{W}_1, \ldots, \boldsymbol{W}_N\}$, $\boldsymbol{W}_i = [\boldsymbol{p}_i \ \boldsymbol{q}_i \ v_i]$, with position $\boldsymbol{p}_i \in \mathbb{R}^3$, unit quaternions $\boldsymbol{q}_i \in \mathbb{S}^3$, and velocity $v_i \in \mathbb{R}$.

5.2.2 3D Joint Tool-Part Affordance Reasoning

To obtain the graspable and functional affordances of tools and parts, we propose an innovative 3D Joint Tool-Part Affordance (JTPA) reasoning model, as shown in Fig. 5.2. Inspired by [113], this model takes the point clouds of both a tool and a part, along with textual affordance labels, and outputs segmented point clouds based on affordance semantics. This process facilitates subsequent task-aware trajectory generation. Within JTPA, we introduce a multi-dimensional fusion module, as shown in Fig. 5.3, which seamlessly integrates semantic and geometric information, thereby augmenting the model's accuracy.

The textual affordance labels are encoded using CLIP's ViT-B/32 [114] text encoder architecture. The processing pipeline comprises:

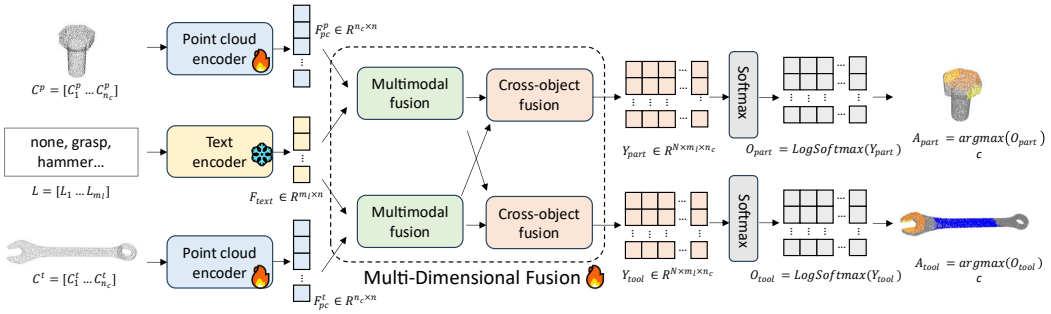


Fig. 5.2: Structure of 3D Joint Tool-Part Affordance (JTPA) reasoning model.

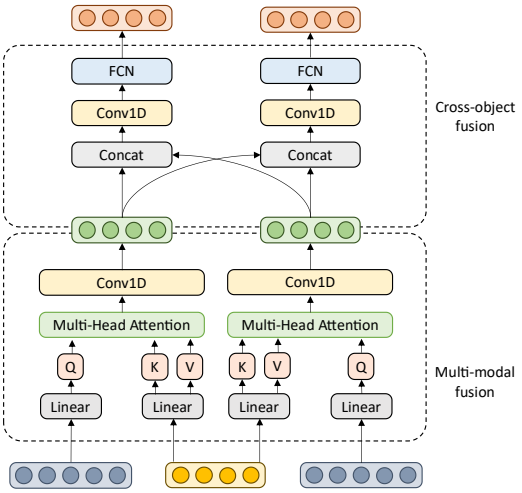


Fig. 5.3: Multi-dimensional fusion module.

1. Tokenization: Input labels are converted to discrete tokens:

$$\mathbf{T} = \text{Tokenize}(L) = [L_1, L_2, \ldots, L_{m_l}] \in \mathbb{Z}^{m_l} \tag{5.1}$$

2. Embedding Projection: Token indices are mapped to continuous vector representations:

$$\mathbf{E} = \text{Embed}(\mathbf{T}) \in \mathbb{R}^{m_l \times d_{\text{embed}}} \tag{5.2}$$

3. Transformer Encoding: The embedded tokens undergo sequential processing through N transformer layers, each implementing:

$$\mathbf{H}_0 = \mathbf{E} + \text{PositionalEncoding}(\mathbf{E}) \tag{5.3}$$

$$\mathbf{H}_l = \text{TransformerLayer}(\mathbf{H}_{l-1}), \quad l = 1, \ldots, N \tag{5.4}$$

where each layer computes:

$$\mathbf{A}_l = \text{MultiHeadAttention}(\mathbf{H}_{l-1}, \mathbf{H}_{l-1}, \mathbf{H}_{l-1}) \tag{5.5}$$

$$\mathbf{H}_l = \text{LayerNorm}(\mathbf{A}_l + \text{FFN}(\text{LayerNorm}(\mathbf{A}_l))) \tag{5.6}$$

4. Feature Projection: The final transformer output is linearly projected to the target dimensionality:

$$\mathbf{F}_{\text{text}} = \text{Proj}(\mathbf{H}_N) \in \mathbb{R}^{m_l \times n} \tag{5.7}$$

This yields per-label semantic embeddings that preserve affordance-specific characteristics.

Geometric features are extracted from tool and part point clouds using PointNet++ [115] backbone with Conv1D and batch norm layers:

1. **PointNet++ Backbone**: Initial geometric features are extracted while preserving point cardinality:

$$\mathbf{F}^t_{\text{init}} = \text{PointNet++}(C^t) \in \mathbb{R}^{n_c \times n} \tag{5.8}$$

$$\mathbf{F}^p_{\text{init}} = \text{PointNet++}(C^p) \in \mathbb{R}^{n_c \times n} \tag{5.9}$$

2. **Feature Refinement**: A 1D convolutional layer with batch normalization enhances local feature discrimination:

$$\mathbf{F}^t_{\text{pc}} = \text{BatchNorm}(\text{Conv1D}(\mathbf{F}^t_{\text{init}})) \in \mathbb{R}^{n_c \times n} \tag{5.10}$$

$$\mathbf{F}^p_{\text{pc}} = \text{BatchNorm}(\text{Conv1D}(\mathbf{F}^p_{\text{init}})) \in \mathbb{R}^{n_c \times n} \tag{5.11}$$

This processing pipeline generates geometrically meaningful per-point embeddings that maintain spatial correspondence with the input point cloud.

To establish semantic alignment between textual affordance labels and geometric point features, a multi-head cross-attention mechanism [116] is employed to enable bidirectional information flow between modalities:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \tag{5.12}$$

$$\mathbf{A}_{\text{tool}} = \text{MultiHead}(\mathbf{Q}_{\text{tool}}, \mathbf{K}_{\text{text}}, \mathbf{V}_{\text{text}}) \tag{5.13}$$

$$\mathbf{A}_{\text{part}} = \text{MultiHead}(\mathbf{Q}_{\text{part}}, \mathbf{K}_{\text{text}}, \mathbf{V}_{\text{text}}) \tag{5.14}$$

where:

- $\mathbf{Q}_{\text{tool}} \in \mathbb{R}^{n_c \times N \times d}$: Query vectors from tool point cloud features
- $\mathbf{Q}_{\text{part}} \in \mathbb{R}^{n_c \times N \times d}$: Query vectors from part point cloud features
- $\mathbf{K}_{\text{text}}, \mathbf{V}_{\text{text}} \in \mathbb{R}^{m_l \times N \times d}$: Key and value vectors from text embeddings
- $d_k = d/\text{num\_heads}$: Dimension per attention head

This configuration ensures each point dynamically aggregates relevant affordance semantics while maintaining computational efficiency. The cross-attention outputs are processed through a 1D convolutional layer for feature refinement:

$$\mathbf{X}_{\text{tool}} = \text{Conv1D}(\mathbf{A}_{\text{tool}}^\top), \quad \mathbf{X}_{\text{part}} = \text{Conv1D}(\mathbf{A}_{\text{part}}^\top) \tag{5.15}$$

This architecture allows each point feature to dynamically attend to relevant textual affordance semantics, creating a geometric-semantic representation space.

To model tool-part affordance interactions, a cross-object fusion that exchanges contextual information between objects is introduced. Affordance representations from both objects are concatenated along the channel dimension:

$$\mathbf{X}_{\text{tool}}^{\text{fused}} = \text{Concat}(\mathbf{X}_{\text{tool}}, \mathbf{X}_{\text{part}}) \in \mathbb{R}^{N \times 2d \times n_c} \tag{5.16}$$

$$\mathbf{X}_{\text{part}}^{\text{fused}} = \text{Concat}(\mathbf{X}_{\text{part}}, \mathbf{X}_{\text{tool}}) \in \mathbb{R}^{N \times 2d \times n_c} \tag{5.17}$$

Then a 1D convolutional layer reduces dimensionality while preserving spatial structure:

$$\mathbf{X}_{\text{tool}}^{\text{red}} = \text{Conv1D}_{2d \to d}(\mathbf{X}_{\text{tool}}^{\text{fused}}) \tag{5.18}$$

$$\mathbf{X}_{\text{part}}^{\text{red}} = \text{Conv1D}_{2d \to d}(\mathbf{X}_{\text{part}}^{\text{fused}}) \tag{5.19}$$

The reduced features are transformed into per-point affordance probabilities through three sequential operations: 1) Feature Flattening: Reshape $\mathbf{X}_{\text{tool}}^{\text{red}} \in \mathbb{R}^{N \times d \times n_c}$ to $\mathbf{X}_{\text{tool}}^{\text{flat}} \in \mathbb{R}^{(N \times n_c) \times d}$; 2) Dimensionality Reduction: Linear projection to affordance space $\mathbf{Y}_{\text{tool}} = \mathbf{W}_{\text{tool}} \mathbf{X}_{\text{tool}}^{\text{flat}} + \mathbf{b}_{\text{tool}} \in \mathbb{R}^{(N \times n_c) \times c}$; 3) Reshaping: Restore spatial dimensions $\mathbf{Y}_{\text{tool}} \in \mathbb{R}^{N \times m_l \times n_c}$.

Final per-point affordance distributions are obtained via log-softmax:

$$\mathbf{O}_{\text{tool}} = \text{LogSoftmax}(\mathbf{Y}_{\text{tool}}), \quad \mathbf{O}_{\text{part}} = \text{LogSoftmax}(\mathbf{Y}_{\text{part}}) \tag{5.20}$$

The predicted affordance label for each point is ultimately determined by selecting the category with the highest log probability from the output distribution:

$$\mathbf{A}_{\text{tool}} = \underset{c}{\text{argmax}}(\mathbf{O}_{\text{tool}}), \quad \mathbf{A}_{\text{part}} = \underset{c}{\text{argmax}}(\mathbf{O}_{\text{part}}), \tag{5.21}$$

where c represents the affordance classes. This operation involves identifying the index of the category with the maximum value for each point, thereby obtaining the predicted label for that point. In this case, the graspable affordance A^g and functional affordance A^f are obtained:

$$\mathbf{A^g} = \underset{c->grasp}{\text{argmax}}(\mathbf{O}_{\text{tool/part}}), \quad \mathbf{A^f} = \underset{c->function}{\text{argmax}}(\mathbf{O}_{\text{tool/part}}). \tag{5.22}$$

In our approach, we directly use the ViT-B/32 text encoder provided by CLIP, which has been pre-trained on a large-scale image-text dataset. This encoder already possesses strong semantic capabilities and can generalize directly to new categories. To avoid overfitting, improve computational efficiency, and enhance training stability, we freeze the parameters of the text encoder during training and only update the parameters of other parts of the JTPA model.

The propose JTPA model is trained end-to-end by optimizing a composite loss function designed to jointly learn tool and part affordances. The total loss L_{total} is a summation of two specialized loss terms: a tool affordance loss L_{tool} and a part affordance loss L_{part}. The loss function is defined as:

$$L_{total} = L_{tool} + L_{part} \tag{5.23}$$

Both components of the loss are formulated as a standard Cross-Entropy (CE) loss, which is a conventional choice for point cloud segmentation tasks. The tool affordance loss, L_{tool}, measures the dissimilarity between the predicted scores for tool classes and the ground-truth tool labels for each point. For a single point cloud containing n_c points, it is given by:

$$L_{tool} = \text{CE}(P_{tool}, Y_{tool}) = -\frac{1}{n_c}\sum_{i=1}^{n_c}\log\left(\frac{\exp(s_{i,c_{tool}})}{\sum_{j=1}^{C_{tool}}\exp(s_{i,j})}\right) \tag{5.24}$$

where:

- n_c is the total number of points in the cloud.
- C_{tool} is the number of possible tool affordance classes.
- $s_{i,j}$ denotes the raw output score (logit) from the network for point i for tool class j.
- c_{tool} is the ground-truth integer label for the tool affordance at point i.

Analogously, the part affordance loss, L_{part}, evaluates the accuracy of the predicted part segmentation against the ground-truth part labels:

$$L_{part} = \text{CE}(P_{part}, Y_{part}) = -\frac{1}{n_c}\sum_{i=1}^{n_c} \log\left(\frac{\exp(s_{i,c_{part}})}{\sum_{j=1}^{C_{part}} \exp(s_{i,j})}\right) \tag{5.25}$$

where:

- C_{part} is the number of part affordance classes.
- $s_{i,j}$ represents the network's logit for point i corresponding to part class j.
- c_{part} is the ground-truth integer label for the part affordance at point i.

By minimizing the combined loss L_{total}, the network learns a shared feature representation that is discriminative for both the tool and part aspects of interaction affordance.

5.2.3 Key Geometric Elements Generation

After obtaining the graspable and functional affordances of each point cloud, geometric features such as centroids, principal axes (handle directions), and contact face normals are extracted using Clustering and Principal Component Analysis (PCA).

Based on functional affordances, the point cloud for each individual tool-part contact face is distinguished by Clustering. Let $\mathbf{P}^f = \{\mathbf{p}_1, ..., \mathbf{p}_n\}$ be the points in functional affordance A^f, the contact face separation is formulated as:

$$\text{Cluster}(\mathbf{P}^f) = \begin{cases} \{\mathbf{P}_1^f, \mathbf{P}_2^f\} & \text{if } \exists \mathbf{n}_1, \mathbf{n}_2 \text{ s.t. } \angle(\mathbf{n}_1, \mathbf{n}_2) > \theta_{\text{threshold}} \\ \{\mathbf{P}^f\} & \text{otherwise} \end{cases} \tag{5.26}$$

where $\mathbf{n}_i$ represents the normal vector of cluster i, and $\theta_{\text{threshold}}$ is the angular threshold for face separation.

Once the point cloud for each segment is available, the central coordinates can be easily obtained by calculating the mean. Subsequently, the direction vectors can be computed using PCA. Let $\mathbf{P} = \{\mathbf{p}_i\}_{i=1}^N$ denote the set of 3D points belonging to a segmented object or part, where each $\mathbf{p}_i \in \mathbb{R}^3$. The geometric centroid $\mathbf{c}$ of the point cloud is:

$$\mathbf{c} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{p}_i \tag{5.27}$$

This centroid represents the spatial center of the object or segment. To determine the principal direction (e.g., the handle direction of a tool), PCA is performed on the corresponding point cloud areas. Then, covariance matrix C can be obtained:

$$C = \frac{1}{N}(X - \mathbf{c})^\top (X - \mathbf{c}) \tag{5.28}$$

where $X \in \mathbb{R}^{N \times 3}$ is the matrix of point coordinates. The dominant direction of the point cloud is described by the eigenvector $\mathbf{v}_1$, which is linked to the maximum eigenvalue λ_1 of C:

$$C\mathbf{v}_1 = \lambda_1 \mathbf{v}_1 \tag{5.29}$$

This vector is normalized to obtain a unit direction:

$$\hat{\mathbf{v}}_1 = \frac{\mathbf{v}_1}{\|\mathbf{v}_1\|} \tag{5.30}$$

For contact face analysis, the segmented point clouds $\mathbf{P}_1$ and $\mathbf{P}_2$ are considered corresponding to two contact faces. Their centroids are given by:

$$\mathbf{c}_1 = \frac{1}{N_1}\sum_{i=1}^{N_1}\mathbf{p}_i^{(1)}, \quad \mathbf{c}_2 = \frac{1}{N_2}\sum_{i=1}^{N_2}\mathbf{p}_i^{(2)} \tag{5.31}$$

The midpoint between the two contact faces, often used as a reference for alignment, is:

$$\mathbf{c}_{\text{area}} = \frac{\mathbf{c}_1 + \mathbf{c}_2}{2} \tag{5.32}$$

To estimate the normal vector of a contact face, PCA is applied to the corresponding face point cloud. The normal $\mathbf{n}$ is taken as the eigenvector associated with the smallest eigenvalue of the covariance matrix C_{face}:

$$C_{\text{face}}\mathbf{n} = \lambda_{\min}\mathbf{n} \tag{5.33}$$

This normal is also normalized to unit length. To ensure consistent orientation, if the dot product between two direction vectors (e.g., handle direction and the vector from the handle to the contact area) is negative, the direction is flipped:

$$\text{if } \mathbf{a}\cdot\mathbf{b} < 0, \quad \mathbf{a} \leftarrow -\mathbf{a} \tag{5.34}$$

For certain tools (e.g., spanners or bolts), the tangent direction between two contact faces is defined as the normalized vector connecting their centroids:

$$\mathbf{t} = \frac{\mathbf{c}_1 - \mathbf{c}_2}{\|\mathbf{c}_1 - \mathbf{c}_2\|} \tag{5.35}$$

This procedure enables robust extraction of geometric features from segmented point clouds, facilitating downstream tasks such as pose estimation, alignment, and manipulation planning.

5.2.4 Task-aware Trajectory Generation

Once the graspable and functional affordances, along with the relevant geometric elements, are established, the LLM can be prompted to integrate these elements to formulate the interim objectives of the task. By incorporating the three-dimensional perception of obstacles and employing conventional trajectory planning methods, the trajectory for the robotic arm can be determined. Different input instructions, even when involving the same tools and parts, necessitate distinct trajectories. For instance, the functional affordance and post-contact trajectory required for using a wrench to tighten an object versus using it to hammer are entirely different. Consequently, we integrate movement affordance to enable the LLM to generate corresponding trajectory code specific to the task, thereby achieving task-aware trajectory generation.

Movement affordance dictates how the tool should be maneuvered to execute the task, as represented by various trajectory shapes, and is closely aligned with functional affordance. When the function is designated as 'hammer,' the movement generally follows a straight downward path, requiring increased speed to achieve the necessary force. This is represented by a straight trajectory, as illustrated in 1. Conversely, when the function is designated as 'screw,' the motion typically entails rotating the tool around a central point at a relatively moderate speed, depicted as a circular trajectory in 2. Each waypoint in the trajectory specifies position, orientation, and speed.

Algorithm 1 Get Straight Trajectory

Input:

$\mathcal{F}_{\text{obs}}$: movable object state function

$\boldsymbol{c}_{\text{voxel}}$: target center in voxel space

$\boldsymbol{\omega}$: unit vector specifying movement direction

d: movement distance in centimeters

v_{in}: desired velocity magnitude (default 1)

Output:

Trajectory waypoints $\mathcal{W} = \{\boldsymbol{W}_1, \ldots, \boldsymbol{W}_N\}$, where $\boldsymbol{W}_i = [\boldsymbol{p}_i \ \boldsymbol{q}_i \ v_i]$, position $\boldsymbol{p}_i \in \mathbb{R}^3$, unit quaternions $\boldsymbol{q}_i \in \mathbb{S}^3$, and velocity $v_i \in \mathbb{R}$

//Initial state

1. $\boldsymbol{p}_0 \leftarrow \mathcal{F}_{\text{obs}}()['\_position\_world'] \ \ \boldsymbol{q}_0 \leftarrow$ end-effector quaternion

//Target calculation

2. $\boldsymbol{c} \leftarrow \text{VoxelToWorld}(\boldsymbol{c}_{\text{voxel}}) \ \ \Delta\boldsymbol{p} \leftarrow \text{cm2index}(d, \boldsymbol{\omega}) \ \ \boldsymbol{c}' \leftarrow \boldsymbol{c} + \Delta\boldsymbol{p}$

//Step setup

3. $\Delta s \leftarrow 0.005$ *//5mm steps*

4. $D \leftarrow \|\boldsymbol{c}' - \boldsymbol{p}_0\| \ \ N \leftarrow \lfloor D/\Delta s \rfloor$

//Initialize

5. $\mathcal{W} \leftarrow \emptyset \ \ \boldsymbol{u}_\omega \leftarrow \boldsymbol{\omega}/\|\boldsymbol{\omega}\|$

//Generate trajectory

6. **for** $n \leftarrow 1$ **to** N **do**

   7. $s_n \leftarrow n \cdot \Delta s$; *//Distance traveled*

   8. $\boldsymbol{p}_n \leftarrow \boldsymbol{p}_0 + s_n \cdot \boldsymbol{u}_\omega$; *//Position along axis*

   9. $\mathcal{W} \leftarrow \mathcal{W} \cup \{[\boldsymbol{p}_n, \boldsymbol{q}_0, v_{\text{in}}]\}$

end

//Final point

10. $\boldsymbol{p}_f \leftarrow \boldsymbol{c}' \ \ \mathcal{W} \leftarrow \mathcal{W} \cup \{[\boldsymbol{p}_f, \boldsymbol{q}_0, v_{\text{in}}]\}$

11. **return** $\mathcal{W}$

Fig. 5.4 illustrates task-aware trajectory code generated by the LLM. In Fig. 5.4(a), when the instruction entails using a spanner to screw a bolt, the

selected functional affordance is 'screw.' The relevant geometric features are then aligned to establish the tool's pose required to contact the bolt before executing the 'screw' action. This alignment involves matching the two segmented surfaces of the tool with those of the bolt while ensuring perpendicularity between the tool and bolt. The 'circle_trajectory' movement affordance is then employed to carry out the 'screw' action. In Fig. 5.4(b), when the instruction involves using a spanner to hammer a bolt, the functional affordance switches to 'hammer.' The associated contact faces are adjusted, requiring the normal of the tool's contact face to be collinear and opposite in direction to the normal of the part's contact face, with an approximate separation of 5 cm. The 'straight_trajectory' movement affordance is then used to perform the hammering action.

Algorithm 2 Get Circle Trajectory

Input:

$\mathcal{F}_{\text{obs}}$: movable object state function

$\boldsymbol{c}_{\text{voxel}}$: circle center in voxel space

$\theta_{\max}$: total rotation angle (default $30°$)

v_{in}: desired velocity (default 1)

Output:

Trajectory waypoints $\mathcal{W} = \{\boldsymbol{W}_1, \ldots, \boldsymbol{W}_N\}$, where $\boldsymbol{W}_i = [\boldsymbol{p}_i \ \boldsymbol{q}_i \ v_i]$, position $\boldsymbol{p}_i \in \mathbb{R}^3$, unit quaternions $\boldsymbol{q}_i \in \mathbb{S}^3$, and velocity $v_i \in \mathbb{R}$

//Initial state

1. $\boldsymbol{p}_0 \leftarrow \mathcal{F}_{\text{obs}}()['\_position\_world'] \ \ \boldsymbol{q}_0 \leftarrow$ end-effector quaternion

//Circle setup

2. $\boldsymbol{c} \leftarrow \text{VoxelToWorld}(\boldsymbol{c}_{\text{voxel}}) \ \ \Delta\theta \leftarrow 2° $ *//Angular step size*

3. $N \leftarrow \lfloor \theta_{\max}/\Delta\theta \rfloor \ \ \boldsymbol{\omega} \leftarrow [0,0,1]^\top$ *//Z-axis rotation*

//Initialize

4. $\mathcal{W} \leftarrow \emptyset \ \ p_z \leftarrow p_{0,z}$ *//Constant z-coordinate*

//Generate circular trajectory

5. **for** $n \leftarrow 1$ **to** N **do**

   6. $\theta_n \leftarrow n \cdot \Delta\theta$ *//Cumulative angle*

   7. $\boldsymbol{\delta} \leftarrow \boldsymbol{p}_{0,xy} - \boldsymbol{c}_{xy}$ *//XY offset*

   8. $\mathbf{R}(\theta_n) \leftarrow \begin{bmatrix} \cos\theta_n & -\sin\theta_n \\ \sin\theta_n & \cos\theta_n \end{bmatrix}$ *//Rotation matrix*

   9. $\boldsymbol{p}_{n,xy} \leftarrow \mathbf{R}(\theta_n)\boldsymbol{\delta} + \boldsymbol{c}_{xy} \ \ \boldsymbol{p}_n \leftarrow [\boldsymbol{p}_{n,x}, \boldsymbol{p}_{n,y}, p_z]^\top$

   10. $\boldsymbol{q}_\Delta \leftarrow \exp(0.5 \cdot [0,0,-\theta_n]^\top)$ *//Delta quaternion*

   11. $\boldsymbol{q}_n \leftarrow \boldsymbol{q}_0 \otimes \boldsymbol{q}_\Delta$

   12. $\mathcal{W} \leftarrow \mathcal{W} \cup \{[\boldsymbol{p}_n, \boldsymbol{q}_n, v_{\text{in}}]\}$

end

//Final point

13. $\boldsymbol{p}_f \leftarrow [\boldsymbol{p}_{N,x}, \boldsymbol{p}_{N,y}, p_z]^\top \ \ \mathcal{W} \leftarrow \mathcal{W} \cup \{[\boldsymbol{p}_f, \boldsymbol{q}_N, v_{\text{in}}]\}$

14. **return** $\mathcal{W}$

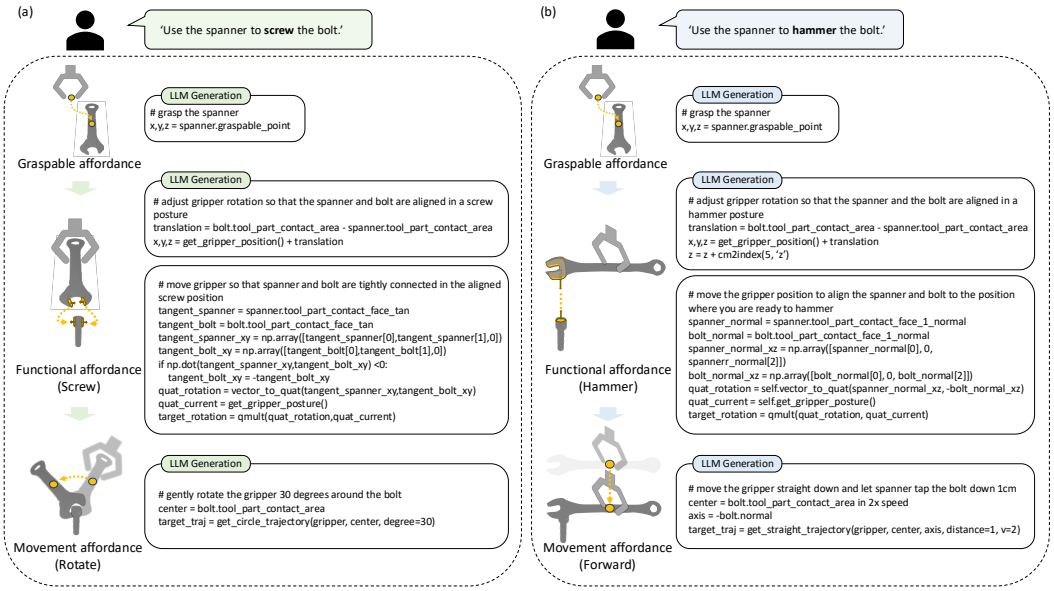


Fig. 5.4: Task-aware trajectory generation.

5.3 Experiments

Comprehensive experiments are conducted to verify the efficiency of the proposed 3D-HiAR approach. First, the ablation experiments are conducted on the JTPA model to verify the effectiveness of the multi-dimensional fusion module. Next, comparative experiments are performed in the simulator to demonstrate the advantages of the proposed hierarchical framework and the role of the three affordances within the entire pipeline. Finally, case studies are provided in real manufacturing settings to validate the proposed method's significant potential for future industrial applications.

5.3.1 Open vocabulary joint tool-part affordance detection

Data collection and model training

The 3D joint tool-part affordance detection model is train on custom dataset including 300 annotated tool and part point clouds (240 for training, 60 for

testing). In this dataset, tools include hammers, spanners, and screwdrivers, while parts include bolts and nails. The textual affordance labels include 'none', 'grasp', 'hammer', and 'screw'. During training, the text encoder remained fixed while the other components of the model were trained in an end-to-end manner. The number of point clouds in both tool and part n_c is fixed to 3000, and the number of extracted features both by point cloud encoder and text encoder n is fixed to 512. The model is trained on a NVIDIA GeForce RTX 3090 GPU.

Affordance detection resutls

The Intersection over Union (IoU) and point accuracy are employed as the evaluation metrics for the point cloud affordance detection experiments. Table 5.1 shows the overall performance of the JTPA model on the custom dataset. The average tool IoU is 0.721 in validation and 0.713 in testing. The average part IoU is 0.668 in validation and 0.664 in testing. The overall accuracy is 0.912 in validation and 0.909 in testing. Overall, the IoU for parts surpasses that for tools due to the more intricate affordance labels associated with tools, which encompass grasp, screw, hammer, and none. In contrast, part segmentation only necessitates reasoning about screw, hammer, and none, without accounting for grasp.

Table 5.1: Affordance Detection Overall Results

Category	Val	Test
Avg. Tool IoU	0.668	0.664
Avg. Part IoU	0.721	0.713
Point Accuracy	0.912	0.909

Table 5.2 provides a more detailed class-wise analysis of the affordance detection results, including the tools, parts, and overall performance within each affordance label. Generally, the evaluation results across the four classes are relatively balanced, with the classes 'none' and 'hammer' achieving higher testing IoU scores of 0.876 and 0.872, respectively. In contrast, 'grasp' and

Table 5.2: Affordance Detection Class-Wise Results

Class	Avg Tool IoU		Avg Part IoU		Overall IoU	
	Val	Test	Val	Test	Val	Test
None	0.692	0.690	0.987	0.986	0.878	0.876
Grasp	0.740	0.728	-	-	0.740	0.728
Screw	0.701	0.677	0.733	0.714	0.720	0.699
Hammer	0.752	0.755	0.954	0.957	0.872	0.872

'screw' performed slightly weaker, with testing IoU scores of 0.728 and 0.699, respectively. For the 'none' class, the IoU for parts significantly exceeds that for tools. This discrepancy arises because parts have only a small portion of point clouds requiring segmentation for functional affordances, with most points classified as 'none.' Conversely, for tools, the majority of points possess graspable or functional affordance point clouds, leaving fewer points for the 'none' category. As a result, parts appear to perform better in this class. For the 'grasp' class, since this class exists only in tools and not in parts, we only calculated the IoU for tools to maintain computational relevancy. Consequently, the overall IoU for this class was directly taken as the average tool IoU. In testing, the IoU for tools in the 'grasp' class reached 0.728. For the 'screw' class, the geometrical surfaces involved are more complex than those of other categories. For example, in a spanner, it consists of two separate planes, and in a flathead screw, it involves a narrow gap. As a result, the 'screw' category yields the lowest results for both tools and parts. However, even with these challenges, the average tool IoU reached 0.677, and the average part IoU reached 0.714, making it a commendable outcome. Compared to 'screw,' the 'hammer' class involves much simpler geometric features, which leads to generally better performance on both tools and parts. The IoU scores reached 0.755 for tools and 0.957 for parts, respectively.

Open vocalulary demonstration

Benefiting from the cross-modal semantic generalization capability of the CLIP text encoder, JTPA achieves robust generalization to novel affordance

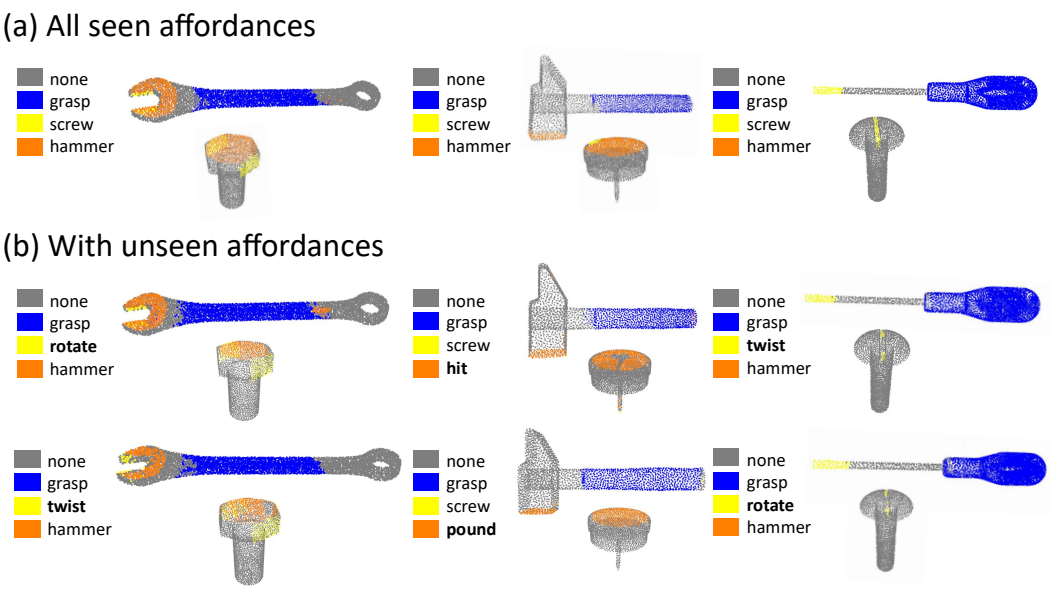


Fig. 5.5: Open vocabulary joint affordance detection demonstration.

labels without task-specific fine-tuning. Fig. 5.5 illustrates the JTPA detection results and open vocabulary demonstrations, where (a) includes seen labels, and (b) includes unseen labels. The affordance detection results are represented in different colors. For the spanner-bolt combination, in (a) the seen labels include 'screw,' while in (b) it is replaced by unseen labels 'rotate' and 'twist.' The yellow-colored areas still correctly highlight the contact regions between the spanner and bolt during the screwing task. For the hammer-nail combination, in (a), the seen label is 'hammer,' shown in orange, and in (b), this label is replaced by 'hit' and 'pound.' JTPA successfully generates the correct contact faces on both the hammer and nail for the hammering task. In the screwdriver and slotted nail combination, the seen label 'screw' is replaced by unseen labels 'twist' and 'rotate.' As shown in (b), JTPA accurately detects and segments the corresponding yellow areas on the screwdriver and nail.

5.3.2 Ablation study

To validate the effectiveness of the proposed multi-dimensional fusion module, ablation experiments are conducted, and the results are presented in

Table 5.3. The structure of JTPA was inspired by the OpenAD model [113], where the point cloud-text fusion module originally relied on correlation. However, cross-attention can dynamically model the global contextual dependencies between the two modalities through learnable attention weights, enabling adaptive feature interaction. Furthermore, cross-object fusion can enhance the geometric feature integration between point clouds, thereby improving the reasoning of functional affordances. Therefore, for the fusion module, we experimented with four different structures: correlation only, cross-attention only, correlation + cross-object fusion, and cross-attention + cross-object fusion. Our experiments revealed that the combination of cross-attention + cross-object fusion achieved the best performance in terms of average tool IoU, average class IoU, and point accuracy. Consequently, the cross-attention + cross-object fusion combination is selected as the multi-dimensional fusion module in JTPA model.

Table 5.3: Affordance Detection Ablation Study Results

Configuration	**Avg. Tool IoU**		**Avg. Part IoU**		**Avg. Class IoU**		**Accuracy**	
	Val	Test	Val	Test	Val	Test	Val	Test
Correlation	0.635	0.627	0.670	0.667	0.755	0.748	0.888	0.884
Cross-Att.	0.576	0.580	0.655	0.640	0.719	0.711	0.872	0.872
Correlation + Cross-Obj.	0.610	0.616	0.622	0.620	0.719	0.719	0.886	0.885
Cross-Att. + Cross-Obj.	**0.721**	**0.713**	0.668	0.664	**0.803**	**0.794**	**0.912**	**0.909**

5.3.3 Comparative experiment

Simulation settings

CoppeliaSim [117] is used as the experimental simulator, with the basic scene settings referencing those from RLBench [118]. Building on this foundation, four task scenarios are created: spanner with bolt (for screwing), spanenr with flat-head nail (for hammering), hammer with flat-head nail (for hammering), and screwdriver with slotted screw (for screwing).

Baseline methods

VoxPoser [51] is one of the most famous methods to combine VLM and LLM for robotic manipulation tasks. The value map framework it introduced effectively integrates the outputs of both models, laying the groundwork for multimodal manipulation. Building on this, MOKA [119] and ReKeP [120] have further refined the representation of object functionality by introducing the concept of keypoint affordance, allowing robots to perform more precise operations on different functional parts of objects. For our baseline, we have chosen VoxPoser, MOKA, and ReKeP to cover the key technological evolution from global value maps to local affordances.

Table 5.4: Comparative Experiment Results

Method	Task	Overall Success Rate	Stage Success Rate			
			Grasp	Contact	Operate	Average
VoxPoser	Screwdriver + Sloted nail (Srewing)	0/10	10/10	0/10	0/10	50.%
	Spanner + Bolt (Screwing)	0/10	10/10	0/10	0/10	
	Spanner + Nail (Hammering)	0/10	10/10	0/10	10/10	
	Hammer + Nail (Hammering)	0/10	10/10	0/10	10/10	
MOKA	Screwdriver + Sloted nail (Srewing)	0/10	10/10	0/10	6/10	75.83%
	Spanner + Bolt (Screwing)	4/10	10/10	3/10	8/10	
	Spanner + Nail (Hammering)	7/10	10/10	7/10	10/10	
	Hammer + Nail (Hammering)	7/10	10/10	7/10	10/10	
ReKeP	Screwdriver + Sloted nail (Srewing)	0/10	10/10	0/10	6/10	78.33%
	Spanner + Bolt (Screwing)	6/10	10/10	3/10	9/10	
	Spanner + Nail (Hammering)	7/10	10/10	8/10	10/10	
	Hammer + Nail (Hammering)	8/10	10/10	8/10	10/10	
3D-HiAR	Screwdriver + Sloted nail (Srewing)	4/10	10/10	6/10	7/10	**90.00%**
	Spanner + Bolt (Screwing)	7/10	10/10	7/10	8/10	
	Spanner + Nail (Hammering)	7/10	10/10	10/10	10/10	
	Hammer + Nail (Hammering)	8/10	10/10	9/10	10/10	

Comparison results

Table 5.4 shows the success rate of 3 baseline models and the proposed 3D-HiAR model in 4 tasks, the 3D-HiAR framework outperforms others by obtaining the best success rate across all tasks, with an overall average success rate of 90.00%. Fig. 5.6 further illustrates the impact of having only object-

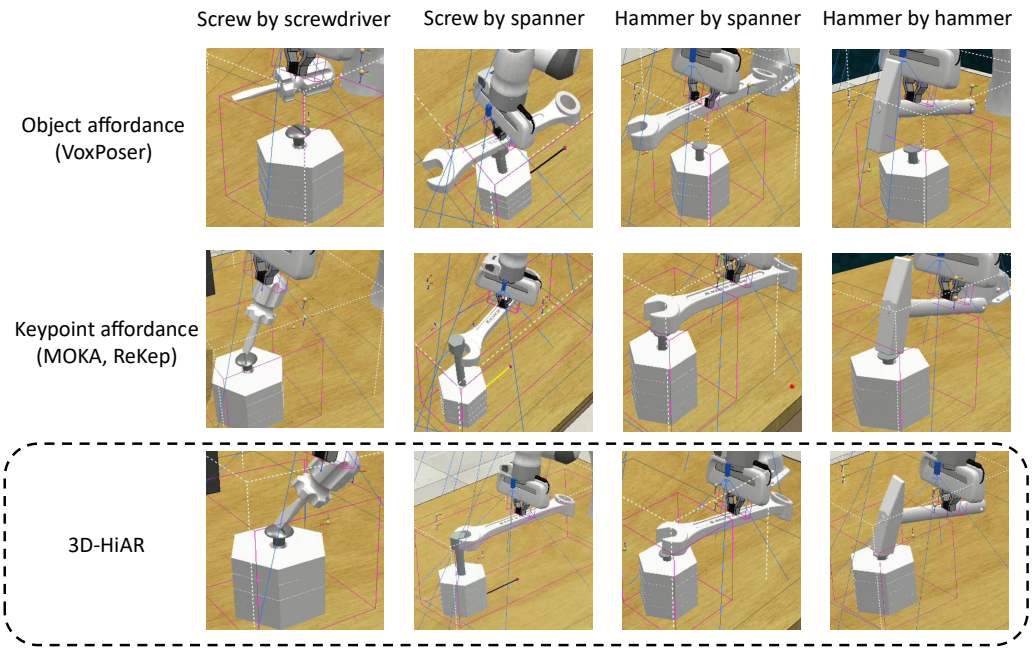


Fig. 5.6: Comparative experiment demonstration.

level affordance, the presence of keypoint affordance, and the proposed 3D-HiAR on the four tasks. For VoxPoser, it only has object-level affordance, where the object is considered the smallest whole. Therefore, for all tasks, the tool is simply picked up and the gripper is moved directly above the center of the part, failing to achieve the correct tool-part contact, which leads to task failure. For the keypoint-based methods like MOKA and REKEP, they refine the affordance of different parts of each object, allowing them to perform the hammering task relatively well. However, due to their reliance solely on keypoint alignment without a more detailed description of geometric relationships (such as surfaces and vectors), they struggle with tasks demanding higher precision, like screwing. This results in difficulties achieving correct tool-part contact, leading to task failure. For our proposed 3D-HiAR, the extraction of geometric elements based on functional affordance significantly improves tool-part contact. Additionally, the subsequent movement affordance allows the system to successfully accomplish task-aware trajectory generation (such as screwing with a spanner and hammering with a spanner). Therefore, we can conclude that our proposed 3D-HiAR effectively aids robots in performing precise tool-use manipulation.

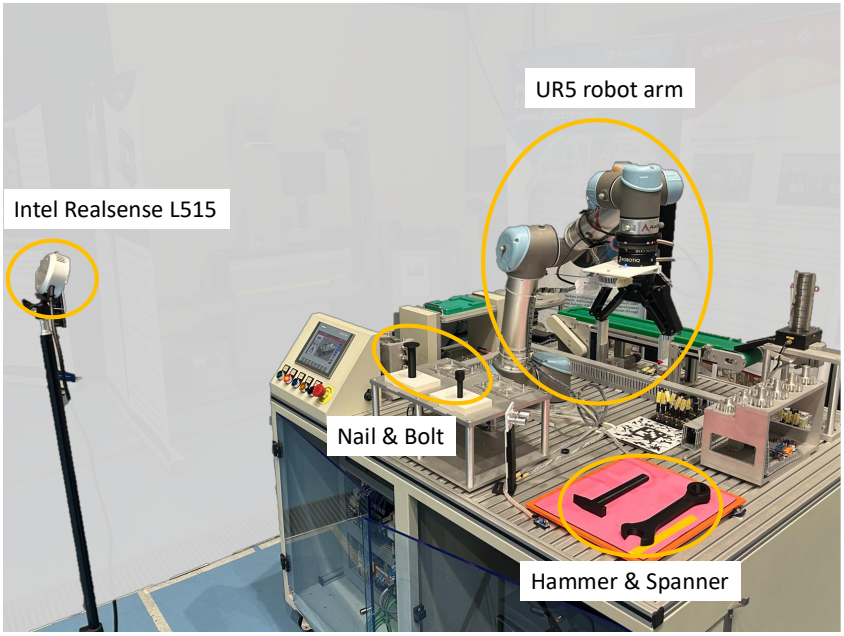


Fig. 5.7: Manufacturing platform setting.

5.3.4 Case study in real manufacturing environment

Real environment setup

Regarding the real-world case study, the proposed 3D-HiAR approach has been deployed in a real manufacturing setting to validate its feasibility in practical industrial applications. As shown in Fig. 5.7, one assembly workstation with UR5 robot arm, one Intel Realsense L515 RGBD camera, the tools (hammer & spanner) and parts (nail & bolt) are included in this platform. Based on this manufacturing platform, two tasks are considered: hammer the nail and screw the bolt.

Manufacturing tasks performance

Fig. 5.8 and Fig. 5.9 present the affordance detection and keyframes during screwing and hammering tasks performance in real manufacturing senario. The success rate for the hammering task is 7/10, and for the screwing task, it

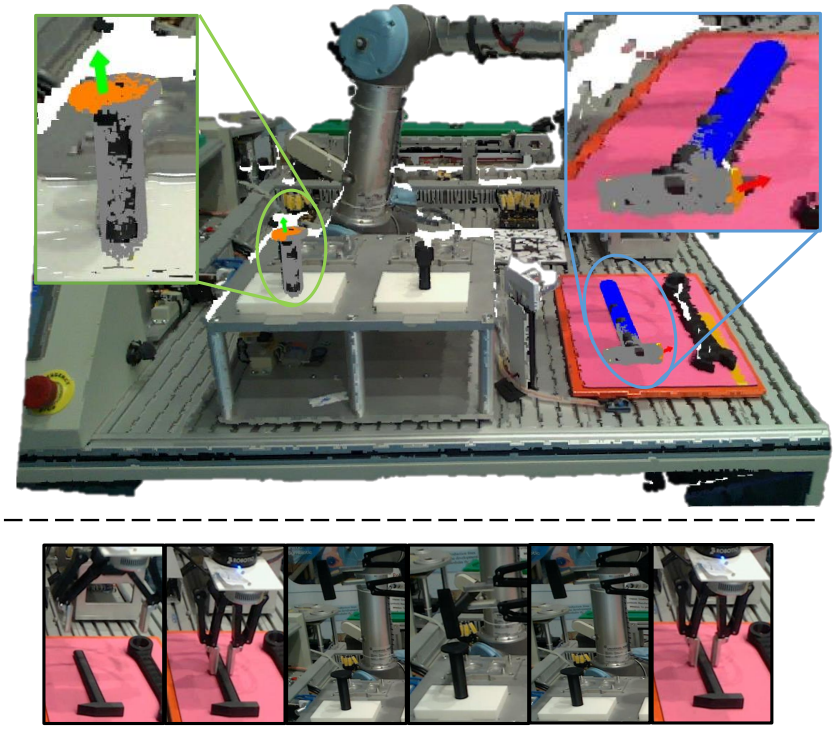


Fig. 5.8: Hammering task in real manufacturing platform.

is 4/10. The main factors affecting task performance include lighting conditions, camera precision, and background interference. Task success rates can be improved by making the background cleaner, ensuring softer lighting, and increasing the number of cameras. Notably, these results are achieved using only a single RGB-D camera without auxiliary sensors, demonstrating robust accuracy under minimal hardware requirements. This efficient configuration makes 3D-HiAR particularly valuable for smart manufacturing, where cost-effective deployment and system simplicity are paramount.

Discussion

In real industrial deployments, the effectiveness of the proposed method depends heavily on the completeness of the acquired point clouds, which in turn imposes stringent requirements on depth-camera accuracy and the post-calibration stability of the camera pose. In simulation, we directly used six cameras to extract point clouds of tools and parts from multiple viewpoints and obtained complete geometry by simply aggregating the six views. In

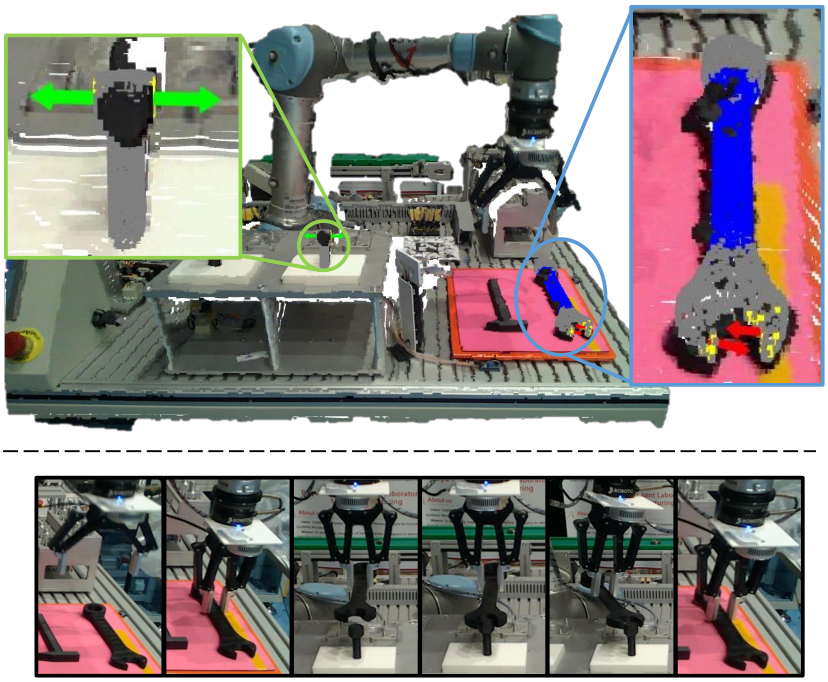


Fig. 5.9: Screwing task in real manufacturing platform.

real-world settings, however, we must first obtain tool and part point clouds via zero-shot object detection and segmentation (GroundingDINO[121] and the Segment Anything Model [122] are considered in our experiments). If multiple cameras are installed at different viewpoints, extensive point-cloud registration is required, and will constrain the feasible task-space size. Thus, only a single camera is implemented in our experiment. In this case, point cloud completion or point cloud 6D pose estimation for aligning with a complete template is required. As zero-shot point cloud completion is prohibitively time-consuming, we instead perform 6D pose estimation on the observed point cloud to coarsely align it with an existing complete point cloud, and then apply point cloud registration for precise refinement, thereby recovering a complete point cloud representation of the scene.

5.4 Summary

This article introduces 3D-HiAR, a foundational model-based method for 3D hierarchical affordance reasoning, concentrating on precise tool-use tasks within industrial settings. This approach comprises two critical components: the first is the 3D Joint Tool-Part Affordance Grounding (JTPA) model, which enables open-vocabulary 3D joint tool-part affordance detection to produce graspable and functional affordances. The second component is Task-aware Trajectory Generation (TaTG), which utilizes an LLM in conjunction with motion affordance to generate motion trajectories for executing intricate tool-use operations. Experimental results demonstrate that the proposed hierarchical affordance reasoning framework surpasses current state-of-the-art methods and is validated through case studies on real manufacturing platforms, highlighting its feasibility in practical industrial scenarios.

The proposed method has some limitations. It is sensitive to the precision of point clouds, necessitating high standards for ambient lighting conditions and camera resolution. Additionally, due to the extended inference chain of LLM and their dependence on network connectivity, trajectory inference is relatively slow. Furthermore, to enhance the success rate of the entire pipeline, the system relies on initial full point cloud alignment or completion, which adds complexity to system deployment. Finally, the proposed VLM exhibits zero-shot capability only in the text modality, and has been evaluated solely on our custom industrial dataset with limited task and scene types.

In the future, there will be a shift towards end-to-end vision-language-action models, integrating 3D affordance detection and the concept of hierarchical reasoning within an end-to-end framework. Furthermore, we plan to incorporate force feedback from the robotic end effector into the model's inference pipeline to execute more complex operational tasks. Additionally, we will consider using humanoid dexterous hands as end effectors to perform tasks

requiring greater flexibility. Finally, we will focus on improving the visual generalization ability, and expanding the diversity of tasks and environments considered.

6 Conclusions

This study demonstrates a significant advancement in human–robot interactive systems for human-centric smart manufacturing, encompassing three fundamental tasks: task planning, navigation, and manipulation. The three corresponding research objectives can be summarized as follows: 1) propose a coarse-to-fine vision-grounded CoT-based task planning method for unforeseen manufacturing scenarios; 2) propose a vision and language-based human guided robotic navigation method in human-centric smart manufacturing with extra consideration of sensor noise problems and free-form interaction manner; 3) propose a vision and language-based 3D hierarchical affordance reasoning approach for industrial robot tool-use manipulation. In this chapter, the key ccorresponding contributions, the limitations and future work will be presented.

6.1 Contributions

This project aims to propose a systematic vision-language model-based methodology for HRI in human-centric manufacturing, in which three aspects are considered: vision and language-multi-robot task planning, vision and language-based mobile robot navigation, and vision and language-based robot arm manipulation. The research works that have been accomplished are listed here:

Contribution 1: Revealed the mechanisms, applications, and significance of multimodal human–robot interaction for human-centric smart manufacturing.

HRI has escalated in notability in recent years, and multimodal communication and control strategies are necessitated to guarantee a secure, efficient, and intelligent HRI experience. In spite of the considerable focus on multimodal HRI, comprehensive disquisitions delineating various modalities and intricately analyzing their combinations remain elusive, consequently limiting holistic understanding and future advancements. This article aspires to bridge this inadequacy by conducting a profound exploration of multimodal HRI, predominantly concentrating on four principal modalities: vision, auditory and language, haptics, and physiological sensing. An extensive review encapsulating algorithmic dissection, interface devices, and applicative dimensions forms part of this discourse. This manuscript distinctively combines multimodal HRI with cognitive science, deeply probing into the three dimensions, perception, cognition, and action, thereby demystifying algorithms intrinsic to multimodal HRI. Finally, it accentuates the empirical challenges and contours of preemptive trajectories for multimodal HRI in human-centric smart manufacturing.

Contribution 2: Proposed a coarse-to-fine vision-grounded chain-of-thought task planning approach for manufacturing unforeseen scenarios.

Task planning plays a crucial role in the whole HRC system, which involves interpreting human instructions and combining them with the current task requirements and environmental conditions to determine the robot's next actions. However, current methods lack integrated global-local contextual reasoning and fail to effectively eliminate issues such as hallucinations and ambiguous expressions in content generated by LLMs and VLMs. To fill the gaps, we proposed a coarse-to-fine vision-grounded chain-of-thought reasoning approach for human-robot interactive task planning for unforeseen scenarios. This approach comprises two essential components: multi-granularity visual parsing and coarse-to-fine CoT reasoning. The multi-granularity visual parsing enables hierarchical segmentation and the generation of mark_ids,

allowing the LLM to reason effectively based on these identifiers and produce clear operational instructions. The coarse-to-fine CoT framework seamlessly integrates global scene understanding with local perspective analysis, facilitating joint reasoning and the generation of stable, executable task plans. The experiment results show the efficiency and innovation of the proposed method, and great significance for unforeseen manufacturing senarios.

Contribution 3: Introduced a vision and language-based human-guided robotic navigation method in unstructured environments for human-centric smart manufacturing.

Within the manufacturing domain, there is a specific scenario where materials and tools are not exclusively positioned on the workstation. This poses a challenge as operators often face disruptions to the manufacturing process due to the need to fetch tools repeatedly. To address this issue, we present an innovative vision and language-based human-guided navigation approach to assist operators in retrieving tools more efficiently. The proposed framework involves various components. Firstly, the 3D point cloud reconstruction and semantic segmentation of a real HRC manufacturing scene are performed, providing a detailed representation of the environment. Secondly, the LLM is leveraged to comprehend natural language commands and generate Python code. Additionally, the Pathfinder module is incorporated for path planning, which aids in optimizing the cobot's navigation towards the desired tools. To evaluate the effectiveness of the proposed framework, the experiment of semantic segmentation, 3D point cloud reconstruction, single-object and multi-object vision and language cobot navigation are conducted in a simulator. Through extensive experimentation, we demonstrate that the AGV within our framework can accurately comprehend complex language instructions and execute the necessary operations to assist human operators in completing manufacturing tasks. These results showcase the potential of our approach in improving operational efficiency and reducing disruptions in the manufacturing process.

Contribution 4: Developed a foundation model-based 3D hierarchical affordance reasoning approach for industrial robot tool-use manipulation.

Robotic manipulation presents a fundamental research challenge in smart manufacturing. Current methodologies primarily concentrate on pick-and-place operations for basic industrial functions such as assembly, material handling, and sorting; however, they exhibit limited proficiency in executing complex tool-use operations, including nail hammering, bolt fastening, or screw driving. To address the challenges, we introduced 3D-HiAR, a foundational model-based 3D Hierarchical Affordance Reasoning approach for robotic tool-use manipulation in industrial tasks. The overall framework comprises two principal components: 1) 3D Joint Tool-Part Affordance (JTPA) reasoning, wherein a novel Vision-Language Model (VLM) with multi-dimensional feature fusion is introduced to jointly infer graspable and functional affordances based on task semantics and the 3D geometric relationships between tools and parts; and 2) Task-aware Trajectory Generation (TaTG), which utilizes an LLM to translate tool-part affordances and linguistic instructions into a series of physical constraints, thus generating post-contact end-effector trajectories as movement affordances. Extensive experiments have demonstrated that the proposed method achieves an overall task success rate of 81.67%, surpassing other current state-of-the-art methods. Additionally, in a real manufacturing environment, we tested the success rates of common industrial tasks with minimal hardware configuration, achieving 70% for the hammering task. This demonstrates the proposed method's great potential significance in future human-centric smart manufacturing.

6.2 Limitations

Limitation 1: The proposed VLM-based HRI approaches largely depend module-based integration and tuning.

Current proposed approaches are primarily modular-based methods, which often involve separately designed and optimized subsystems for perception, planning, and control. While such decomposition offers interpretability and ease of debugging, it may lead to suboptimal performance due to error accumulation between modules and limited ability to generalize to unseen scenarios. Furthermore, modular systems often require significant manual engineering and domain-specific tuning, reducing their scalability and adaptability in dynamic real-world environments.

Limitation 2: The proposed methodology only considers navigation and manipulation as separate processes.

Current systems typically address manipulation and navigation as independent tasks, employing decoupled policies for each capability. This division often results in inefficient and uncoordinated behaviors when robots operate in scenarios requiring simultaneous mobility and interaction, such as moving while handling objects or adjusting posture to reach a target. The lack of integrated control leads to suboptimal performance, increased task completion time, and limited applicability in dynamic human-shared environments.

Limitation 3: The input and output modalities are only restricted to vision and language.

Existing methods predominantly rely on vision and language modalities, overlooking the critical role of physical interactions, such as force and torque feedback. This omission limits the robot's ability to perform tasks requiring precise contact, compliance, or delicate manipulation—e.g., assembly, insertion, or handling fragile objects. As a result, current systems may lack robustness in real-world scenarios where tactile and haptic cues are essential for successful and safe execution.

6.3 Future work

Future work 1: Towards end-to-end vision-language-action models for robot manipulation and navigation.

To overcome the limitations of current modular systems, a promising direction is the development of end-to-end Vision-Language-Action (VLA) models. Such models would integrate perception, reasoning, and control into a unified neural architecture, trained jointly to translate natural language instructions and visual observations directly into robot actions. By bypassing hand-designed intermediate representations and avoiding error propagation across separate modules, end-to-end VLAs could achieve stronger generalization, improved robustness, and greater flexibility in open-world human-centric environments, such as smart manufacturing facilities. Future work should focus on collecting large-scale multimodal task demonstrations, designing effective architectures for real-time control, and incorporating safety-aware learning mechanisms to enable reliable and scalable deployment in dynamic settings.

Future work 2: Unified Learning for Loco-manipulation.

A promising future direction is to develop unified learning frameworks for loco-manipulation, which jointly optimize navigation and manipulation policies within a single model. Such systems could dynamically coordinate motion and interaction strategies in real time, enabling seamless task execution in complex industrial settings. Future efforts should focus on multi-task reinforcement learning, shared representation learning across modalities, and sim-to-real transfer to facilitate robust and efficient loco-manipulation capabilities.

Future work 3: Integration of force-torque sensing for multimodal learning.

Future research will explore the integration of force-torque sensing to develop multimodal vision-language-force models. By incorporating haptic feedback, robots can achieve more nuanced and adaptive control in contact-rich tasks. This direction involves designing fusion mechanisms for cross-modal reasoning, collecting multimodal datasets with physical interactions, and developing compliant control policies that leverage both perceptual and physical feedback for safer and more precise manipulation in unstructured environments.

References

[1] Baicun Wang, Pai Zheng, Yue Yin, Albert Shih, and Lihui Wang. “Toward human-centric smart manufacturing: A human-cyber-physical systems (HCPS) perspective”. In: *Journal of Manufacturing Systems* 63 (2022), pp. 471–490 (cit. on pp. 4, 12).

[2] Jiewu Leng, Weinan Sha, Baicun Wang, et al. “Industry 5.0: Prospect and retrospect”. In: *Journal of Manufacturing Systems* 65 (2022), pp. 279–295 (cit. on pp. 4, 12).

[3] Gbenga Abiodun Odesanmi, Qining Wang, and Jingeng Mai. “Skill learning framework for human–robot interaction and manipulation tasks”. In: *Robotics and Computer-Integrated Manufacturing* 79 (2023), p. 102444 (cit. on pp. 4, 11, 13).

[4] Zhaojun Qin and Yuqian Lu. “A Knowledge Graph-based knowledge representation for adaptive manufacturing control under mass personalization”. In: *Manufacturing Letters* 35 (2023), pp. 96–104 (cit. on p. 4).

[5] Jiachen Li, Fan Yang, Hengbo Ma, et al. “Rain: Reinforced hybrid attention inference network for motion forecasting”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 16096–16106 (cit. on p. 4).

[6] Roy S Hessels, Martin K Teunisse, Diederick C Niehorster, et al. “Task-related gaze behaviour in face-to-face dyadic collaboration: Toward an interactive theory?” In: *Visual Cognition* 31.4 (2023), pp. 291–313 (cit. on p. 4).

[7] Adam Fishman, Chris Paxton, Wei Yang, et al. “Collaborative interaction models for optimized human-robot teamwork”. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 11221–11228 (cit. on p. 4).

[8] Mohammad Samin Yasar and Tariq Iqbal. “A scalable approach to predict multi-agent motion for human-robot collaboration”. In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 1686–1693 (cit. on p. 4).

[9] Shukang Yin, Chaoyou Fu, Sirui Zhao, et al. “A Survey on Multimodal Large Language Models”. In: *arXiv preprint arXiv:2306.13549* (2023) (cit. on pp. 5, 58).

[10] Luca Ragno, Alberto Borboni, Federica Vannetti, Cinzia Amici, and Nicoletta Cusano. “Application of Social Robots in Healthcare: Review on Characteristics, Requirements, Technical Solutions”. In: *Sensors* 23.15 (2023), p. 6820 (cit. on p. 11).

[11] Vincent Wing Sun Tung and Rob Law. “The potential for tourism and hospitality experience research in human-robot interactions”. In: *International Journal of Contemporary Hospitality Management* 29.10 (2017), pp. 2498–2513 (cit. on p. 11).

[12] Olatunji Mumini Omisore, Shipeng Han, Jing Xiong, et al. “A review on flexible robotic systems for minimally invasive surgery”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 52.1 (2020), pp. 631–644 (cit. on p. 11).

[13] Ming Zhang, Rui Xu, Haitao Wu, Jia Pan, and Xiaowei Luo. “Human–robot collaboration for on-site construction”. In: *Automation in Construction* 150 (2023), p. 104812 (cit. on p. 11).

[14] Lefteris Benos, Vasileios Moysiadis, Dimitrios Kateris, et al. “Human–Robot Interaction in Agriculture: A Systematic Review”. In: *Sensors* 23.15 (2023), p. 6776 (cit. on p. 12).

[15] H James Wilson and Paul R Daugherty. “Collaborative intelligence: Humans and AI are joining forces”. In: *Harvard Business Review* 96.4 (2018), pp. 114–123 (cit. on p. 12).

[16] Lihui Wang, R Gao, József Váncza, et al. “Symbiotic human-robot collaborative assembly”. In: *CIRP annals* 68.2 (2019), pp. 701–726 (cit. on pp. 12, 31, 58, 82).

[17] Sehoon Kim. “Working With Robots: Human Resource Development Considerations in Human–Robot Interaction”. In: *Human Resource Development Review* 21.1 (2022), pp. 48–74 (cit. on p. 13).

[18] Haluk Ogmen, Kazuhisa Shibata, and Arash Yazdanbakhsh. “Perception, cognition, and action in hyperspaces: Implications on brain plasticity, learning, and cognition”. In: *Frontiers in Psychology* (2020), p. 3000 (cit. on p. 13).

[19] Christos Gkournelos, Christos Konstantinou, and Sotiris Makris. “An LLM-based approach for enabling seamless Human-Robot collaboration in assembly”. In: *CIRP Annals* 73.1 (2024), pp. 9–12 (cit. on pp. 16, 17, 33).

[20] Yue Yin, Ke Wan, Chengxi Li, and Pai Zheng. “An LLM-enabled human demonstration-assisted hybrid robot skill synthesis approach for human-robot collaborative assembly”. In: *CIRP Annals* (2025) (cit. on pp. 16, 17, 32, 82).

[21] Fanru Gao, Liqiao Xia, Jianjing Zhang, et al. “Integrating large language model for natural language-based instruction toward robust human-robot collaboration”. In: *Procedia CIRP* 130 (2024), pp. 313–318 (cit. on pp. 16, 17, 33).

[22] Sichao Liu, Jianjing Zhang, Lihui Wang, and Robert X Gao. “Vision AI-based human-robot collaborative assembly driven by autonomous robots”. In: *CIRP annals* 73.1 (2024), pp. 13–16 (cit. on pp. 16, 17, 20, 22, 32, 59).

[23] Junming Fan and Pai Zheng. “A vision-language-guided robotic action planning approach for ambiguity mitigation in human–robot collaborative manufacturing”. In: *Journal of Manufacturing Systems* 74 (2024), pp. 1009–1018 (cit. on pp. 16, 17, 25, 32).

[24] Yiwei Hua, Kerun Li, Ru Wang, et al. “Integration of dynamic knowledge and LLM for adaptive human-robot collaborative assembly solution generation”. In: *Advanced Engineering Informatics* 68 (2025), p. 103613 (cit. on pp. 16, 17, 33).

[25] Nadun Ranasinghe, Wael M Mohammed, Kostas Stefanidis, and Jose L Martinez Lastra. “Large Language Models in Human-Robot Collaboration with Cognitive Validation Against Context-induced Hallucinations”. In: *IEEE Access* (2025) (cit. on pp. 16, 18, 33).

[26] Silvia Izquierdo-Badiola, Gerard Canal, Carlos Rizzo, and Guillem Alenyà. “Plancollabnl: Leveraging large language models for adaptive plan generation in human-robot collaboration”. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2024, pp. 17344–17350 (cit. on pp. 16, 18, 33).

[27] Tian Wang, Junming Fan, Pai Zheng, Ruqiang Yan, and Lihui Wang. “Vision-Language Model-Based Human-Guided Mobile Robot Navigation in an Unstructured Environment for Human-Centric Smart Manufacturing”. In: *Engineering* (2025) (cit. on pp. 16, 17, 32).

[28] Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. “Simple but effective: Clip embeddings for embodied ai”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 14829–14838 (cit. on pp. 19, 20).

[29] Ryosuke Korekata, Motonari Kambara, Yu Yoshida, et al. “Switching head-tail funnel UNITER for dual referring expression comprehension with fetch-and-carry tasks”. In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2023, pp. 3865–3872 (cit. on pp. 20, 21).

[30] Yicong Hong, Yang Zhou, Ruiyi Zhang, et al. “Learning navigational visual representations with semantic map supervision”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 3055–3067 (cit. on pp. 20, 21).

[31] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. “Visual language maps for robot navigation”. In: *arXiv preprint arXiv:2210.05714* (2022) (cit. on pp. 20, 21, 74, 81).

[32] Yanyuan Qiao, Yuankai Qi, Zheng Yu, Jing Liu, and Qi Wu. “March in chat: Interactive prompting for remote embodied referring expression”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2023, pp. 15758–15767 (cit. on pp. 20, 21).

[33] Chen Gao, Si Liu, Jinyu Chen, et al. “Room-object entity prompting and reasoning for embodied referring expression”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.2 (2023), pp. 994–1010 (cit. on pp. 20, 21).

[34] Gengze Zhou, Yicong Hong, and Qi Wu. “Navgpt: Explicit reasoning in vision-and-language navigation with large language models”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 7. 2024, pp. 7641–7649 (cit. on pp. 20, 22).

[35] Xiaohan Wang, Wenguan Wang, Jiayi Shao, and Yi Yang. “Learning to follow and generate instructions for language-capable navigation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.5 (2023), pp. 3334–3350 (cit. on pp. 20, 21).

[36] Bahram Mohammadi, Yicong Hong, Yuankai Qi, et al. “Augmented commonsense knowledge for remote object grounding”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 38. 5. 2024, pp. 4269–4277 (cit. on pp. 20, 21).

[37] Bingqian Lin, Yunshuang Nie, Ziming Wei, et al. “Correctable landmark discovery via large models for vision-language navigation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.12 (2024), pp. 8534–8548 (cit. on pp. 20, 22).

[38] Jiawei Wang, Teng Wang, Lele Xu, Zichen He, and Changyin Sun. “Discovering intrinsic subgoals for vision-and-language navigation via hierarchical reinforcement learning”. In: *IEEE Transactions on Neural Networks and Learning Systems* 36.4 (2024), pp. 6516–6528 (cit. on pp. 20, 22).

[39] Tian Wang, Junming Fan, and Pai Zheng. “An LLM-based vision and language cobot navigation approach for Human-centric Smart Manufacturing”. In: *Journal of Manufacturing Systems* (2024) (cit. on pp. 20, 23, 59, 82, 83).

[40] Dhruv Shah, Błażej Osiński, Sergey Levine, et al. “Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action”. In: *Conference on Robot Learning*. PMLR. 2023, pp. 492–504 (cit. on pp. 20, 23).

[41] Raphael Schumann, Wanrong Zhu, Weixi Feng, et al. “Velma: Verbalization embodiment of llm agents for vision and language navigation in street view”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 17. 2024, pp. 18924–18933 (cit. on pp. 20, 23).

[42] Jialu Li, Aishwarya Padmakumar, Gaurav Sukhatme, and Mohit Bansal. “Vln-video: Utilizing driving videos for outdoor vision-and-language navigation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 17. 2024, pp. 18517–18526 (cit. on pp. 20, 23).

[43] Walter Goodwin, Sagar Vaze, Ioannis Havoutis, and Ingmar Posner. “Semantically grounded object matching for robust robotic scene rearrangement”. In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 11138–11144 (cit. on pp. 25–27).

[44] Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. “Act3d: 3d feature field transformers for multi-task robotic manipulation”. In: *arXiv preprint arXiv:2306.17817* (2023) (cit. on pp. 25, 27).

[45] Youngsun Wi, Mark Van der Merwe, Pete Florence, Andy Zeng, and Nima Fazeli. “CALAMARI: Contact-aware and language conditioned spatial action MApping for contact-RIch manipulation”. In: *7th Annual Conference on Robot Learning*. 2023 (cit. on pp. 25, 27).

[46] Yanjie Ze, Ge Yan, Yueh-Hua Wu, et al. “Gnfactor: Multi-task real robot learning with generalizable neural feature fields”. In: *Conference on robot learning*. PMLR. 2023, pp. 284–301 (cit. on p. 25).

[47] Junghyun Kim, Gi-Cheon Kang, Jaein Kim, Suyeon Shin, and Byoung-Tak Zhang. “Gvcci: Lifelong learning of visual grounding for language-guided robotic manipulation”. In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2023, pp. 952–959 (cit. on p. 25).

[48] Austin Stone, Ted Xiao, Yao Lu, et al. “Open-world object manipulation using pre-trained vision-language models”. In: *arXiv preprint arXiv:2303.00905* (2023) (cit. on p. 25).

[49] Danny Driess, Fei Xia, Mehdi SM Sajjadi, et al. “Palm-e: An embodied multimodal language model”. In: (2023) (cit. on pp. 25, 26).

[50] Satvik Sharma, Huang Huang, Kaushik Shivakumar, et al. “Semantic mechanical search with large vision and language models”. In: *arXiv preprint arXiv:2302.12915* (2023) (cit. on pp. 25, 26).

[51] Wenlong Huang, Chen Wang, Ruohan Zhang, et al. “Voxposer: Composable 3d value maps for robotic manipulation with language models”. In: *arXiv preprint arXiv:2307.05973* (2023) (cit. on pp. 25–27, 104).

[52] Xun Xu, Yuqian Lu, Birgit Vogel-Heuser, and Lihui Wang. “Industry 4.0 and Industry 5.0—Inception, conception and perception”. In: *Journal of manufacturing systems* 61 (2021), pp. 530–535 (cit. on pp. 31, 57, 82).

[53] Gilde Vanel Tchane Djogdom, Ramy Meziane, and Martin J-D Otis. “Robust dynamic robot scheduling for collaborating with humans in manufacturing operations”. In: *Robotics and Computer-Integrated Manufacturing* 88 (2024), p. 102734 (cit. on p. 31).

[54] Anna Presciuttini, Alessandra Cantini, Federica Costa, and Alberto Portioli-Staudacher. “Machine learning applications on IoT data in manufacturing operations and their interpretability implications: A systematic literature review”. In: *Journal of Manufacturing Systems* 74 (2024), pp. 477–486 (cit. on p. 31).

[55] Lei Ren, Jiabao Dong, Shuai Liu, Lin Zhang, and Lihui Wang. “Embodied intelligence toward future smart manufacturing in the era of AI foundation model”. In: *IEEE/ASME Transactions on Mechatronics* (2024) (cit. on p. 32).

[56] Zhigen Zhao, Shuo Cheng, Yan Ding, et al. “A survey of optimization-based task and motion planning: From classical to learning approaches”. In: *IEEE/ASME Transactions on Mechatronics* (2024) (cit. on p. 32).

[57] Julio C Serrano-Ruiz, Josefa Mula, and Raúl Poler. “Smart manufacturing scheduling: A literature review”. In: *Journal of Manufacturing Systems* 61 (2021), pp. 265–287 (cit. on p. 32).

[58] Jonghan Lim, Birgit Vogel-Heuser, and Ilya Kovalenko. “Large language model-enabled multi-agent manufacturing systems”. In: *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*. IEEE. 2024, pp. 3940–3946 (cit. on p. 32).

[59] Han Wang, Chenze Wang, Qing Liu, et al. “A data and knowledge driven autonomous intelligent manufacturing system for intelligent factories”. In: *Journal of Manufacturing Systems* 74 (2024), pp. 512–526 (cit. on p. 32).

[60] Vishal Pallagani, Bharath Chandra Muppasani, Kaushik Roy, et al. “On the prospects of incorporating large language models (llms) in automated planning and scheduling (aps)”. In: *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 34. 2024, pp. 432–444 (cit. on p. 32).

[61] Junming Fan, Yue Yin, Tian Wang, et al. “Vision-language model-based human-robot collaboration for smart manufacturing: A state-of-the-art survey”. In: *Frontiers of Engineering Management* 12.1 (2025), pp. 177–200 (cit. on pp. 32, 82).

[62] Junda He, Christoph Treude, and David Lo. “LLM-Based Multi-Agent Systems for Software Engineering: Literature Review, Vision, and the Road Ahead”. In: *ACM Transactions on Software Engineering and Methodology* 34.5 (2025), pp. 1–30 (cit. on p. 32).

[63] Jianwei Yang, Hao Zhang, Feng Li, et al. “Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v”. In: *arXiv preprint arXiv:2310.11441* (2023) (cit. on p. 38).

[64] Ze Liu, Yutong Lin, Yue Cao, et al. “Swin transformer: Hierarchical vision transformer using shifted windows”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 10012–10022 (cit. on p. 38).

[65] Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. “Vision transformer with deformable attention”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 4794–4803 (cit. on p. 38).

[66] Feng Li, Hao Zhang, Huaizhe Xu, et al. “Mask dino: Towards a unified transformer-based framework for object detection and segmentation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 3041–3050 (cit. on p. 38).

[67] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, et al. “Feature pyramid networks for object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125 (cit. on p. 39).

[68] Aaron Hurst, Adam Lerer, Adam P Goucher, et al. “Gpt-4o system card”. In: *arXiv preprint arXiv:2410.21276* (2024) (cit. on p. 50).

[69] Jinshan Wang, Bo Tao, Zeyu Gong, Supu Yu, and Zhouping Yin. “A mobile robotic measurement system for large-scale complex components based on optical scanning and visual tracking”. In: *Robotics and Computer-Integrated Manufacturing* 67 (2021), p. 102010 (cit. on p. 57).

[70] Abdurrahman Yilmaz, Ecem Sumer, and Hakan Temeltas. “A precise scan matching based localization method for an autonomously guided vehicle in smart factories”. In: *Robotics and Computer-Integrated Manufacturing* 75 (2022), p. 102302 (cit. on pp. 57, 59).

[71] Jie Meng, Shuting Wang, Yuanlong Xie, et al. “A safe and efficient LIDAR-based navigation system for 4WS4WD mobile manipulators in manufacturing plants”. In: *Measurement Science and Technology* 32.4 (2021), p. 045203 (cit. on p. 57).

[72] Pai Zheng, Chengxi Li, Junming Fan, and Lihui Wang. “A vision-language-guided and deep reinforcement learning-enabled approach for unstructured human-robot collaborative manufacturing task fulfilment”. In: *CIRP Annals* (2024) (cit. on p. 57).

[73] Emanuela Del Dottore, Alessio Mondini, Nick Rowe, and Barbara Mazzolai. "A growing soft robot with climbing plant–inspired adaptive behaviors for navigation in unstructured environments". In: *Science Robotics* 9.86 (2024), eadi5908 (cit. on p. 58).

[74] Liuao Pei, Junxiao Lin, Zhichao Han, et al. "Collaborative Planning for Catching and Transporting Objects in Unstructured Environments". In: *IEEE Robotics and Automation Letters* (2023) (cit. on p. 58).

[75] Pai Zheng, Shufei Li, Junming Fan, Chengxi Li, and Lihui Wang. "A collaborative intelligence-based approach for handling human-robot collaboration uncertainties". In: *CIRP annals* 72.1 (2023), pp. 1–4 (cit. on p. 58).

[76] Pai Zheng, Shufei Li, Liqiao Xia, Lihui Wang, and Aydin Nassehi. "A visual reasoning-based approach for mutual-cognitive human-robot collaboration". In: *CIRP annals* 71.1 (2022), pp. 377–380 (cit. on p. 58).

[77] Shufei Li, Pai Zheng, Sichao Liu, et al. "Proactive human–robot collaboration: Mutual-cognitive, predictable, and self-organising perspectives". In: *Robotics and Computer-Integrated Manufacturing* 81 (2023), p. 102510 (cit. on p. 58).

[78] Junming Fan, Pai Zheng, and Shufei Li. "Vision-based holistic scene understanding towards proactive human–robot collaboration". In: *Robotics and Computer-Integrated Manufacturing* 75 (2022), p. 102304 (cit. on p. 58).

[79] Tian Wang, Pai Zheng, Shufei Li, and Lihui Wang. "Multimodal Human–Robot Interaction for Human-Centric Smart Manufacturing: A Survey". In: *Advanced Intelligent Systems* (2023), p. 2300359 (cit. on p. 58).

[80] Mengyang Ren and Pai Zheng. "Towards smart product-service systems 2.0: A retrospect and prospect". In: *Advanced Engineering Informatics* 61 (2024), p. 102466 (cit. on pp. 58, 83).

[81] Liqiao Xia, Chengxi Li, Canbin Zhang, Shimin Liu, and Pai Zheng. "Leveraging error-assisted fine-tuning large language models for manufacturing excellence". In: *Robotics and Computer-Integrated Manufacturing* 88 (2024), p. 102728 (cit. on p. 58).

[82] Mengyang Ren, Liang Dong, Ziqing Xia, Jingchen Cong, and Pai Zheng. "A proactive interaction design method for personalized user context prediction in smart-product service system". In: *Procedia CIRP* 119 (2023), pp. 963–968 (cit. on pp. 58, 83).

[83] Shijian Su, Xianping Zeng, Shuang Song, et al. "Positioning accuracy improvement of automated guided vehicles based on a novel magnetic tracking approach". In: *IEEE Intelligent Transportation Systems Magazine* 12.4 (2018), pp. 138–148 (cit. on p. 59).

[84] Mithun Goutham, Stephen Boyle, Meghna Menon, et al. “Optimal path planning through a sequence of waypoints”. In: *IEEE Robotics and Automation Letters* 8.3 (2023), pp. 1509–1514 (cit. on p. 59).

[85] Guillaume Demesure, Hind Bril El-Haouzi, and Benoit Iung. “Mobile-agents based hybrid control architecture—implementation of consensus algorithm in hierarchical control mode”. In: *CIRP Annals* 70.1 (2021), pp. 385–388 (cit. on p. 59).

[86] Alexander Pashevich, Cordelia Schmid, and Chen Sun. “Episodic transformer for vision-and-language navigation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 15942–15952 (cit. on p. 59).

[87] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. “Visual language maps for robot navigation”. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 10608–10615 (cit. on p. 59).

[88] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. “Robust reconstruction of indoor scenes”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 5556–5565 (cit. on pp. 62, 63).

[89] Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. “Colored point cloud registration revisited”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 143–152 (cit. on p. 63).

[90] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. “Fast global registration”. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer. 2016, pp. 766–782 (cit. on p. 63).

[91] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, et al. “Kinectfusion: Real-time dense surface mapping and tracking”. In: *2011 10th IEEE international symposium on mixed and augmented reality*. Ieee. 2011, pp. 127–136 (cit. on p. 63).

[92] Szymon Rusinkiewicz and Marc Levoy. “Efficient variants of the ICP algorithm”. In: *Proceedings third international conference on 3-D digital imaging and modeling*. IEEE. 2001, pp. 145–152 (cit. on p. 63).

[93] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, et al. “Habitat: A platform for embodied ai research”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 9339–9347 (cit. on pp. 64, 75).

[94] Alexander Brodén and Gustav Pihl Bohlin. *Towards real-time navmesh generation using gpu accelerated scene voxelization*. 2017 (cit. on p. 64).

[95] Wang Wenna, Ding Weili, Hua Changchun, et al. “A digital twin for 3D path planning of large-span curved-arm gantry robot”. In: *Robotics and Computer-Integrated Manufacturing* 76 (2022), p. 102330 (cit. on p. 64).

[96] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. "Language-driven Semantic Segmentation". In: *International Conference on Learning Representations*. 2022 (cit. on pp. 64, 69, 70).

[97] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer. 2015, pp. 234–241 (cit. on p. 69).

[98] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. "Encoder-decoder with atrous separable convolution for semantic image segmentation". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 801–818 (cit. on p. 69).

[99] Pai Zheng, Yue Yin, Tian Wang, and Ke Wan. "Society 5.0: Social implications, technoethics, and social acceptance". In: *Manufacturing from Industry 4.0 to Industry 5.0*. Elsevier, 2024, pp. 133–178 (cit. on p. 82).

[100] Baicun Wang, Pai Zheng, Ci Song, Dimitris Mourtzis, and Lihui Wang. "Future Research Directions on Human-Centric Smart Manufacturing". In: *Human-Centric Smart Manufacturing Towards Industry 5.0* (2025), pp. 359–369 (cit. on p. 82).

[101] Wenhang Dong, Shufei Li, and Pai Zheng. "Toward Embodied Intelligence-Enabled Human–Robot Symbiotic Manufacturing: A Large Language Model-Based Perspective". In: *Journal of Computing and Information Science in Engineering* 25.5 (2025) (cit. on p. 82).

[102] Tian Wang, Pai Zheng, Shufei Li, and Lihui Wang. "Multimodal human–robot interaction for human-centric smart manufacturing: a survey". In: *Advanced Intelligent Systems* 6.3 (2024), p. 2300359 (cit. on p. 82).

[103] Benhua Gao, Junming Fan, and Pai Zheng. "Empower dexterous robotic hand for human-centric smart manufacturing: A perception and skill learning perspective". In: *Robotics and Computer-Integrated Manufacturing* 93 (2025), p. 102909 (cit. on p. 82).

[104] LIU Peifeng, Lu Qian, Xingwei Zhao, and Bo Tao. "Joint knowledge graph and large language model for fault diagnosis and its application in aviation assembly". In: *IEEE Transactions on Industrial Informatics* 20.6 (2024), pp. 8160–8169 (cit. on p. 82).

[105] Devis Bartsch, Christian Borck, Martin Behm, and Jacob Böhnke. "Development of a Multi-layered Quality Assurance Framework for Manual Assembly Processes in the Aviation Industry". In: *Procedia CIRP* 130 (2024), pp. 1227–1233 (cit. on p. 82).

[106] David Blanco, Eva María Rubio, Beatriz Agustina, Marta María Marín, and Ana María Camacho. “Advanced procedures and scheduling for aircraft assembly processes: A systematic review approach”. In: *Procedia CIRP* 126 (2024), pp. 817–822 (cit. on p. 82).

[107] Zhenyu Lu, Ning Wang, and Chenguang Yang. “A dynamic movement primitives-based tool use skill learning and transfer framework for robot manipulation”. In: *IEEE Transactions on Automation Science and Engineering* (2024) (cit. on p. 83).

[108] Yicong Li, Na Zhao, Junbin Xiao, et al. “Laso: Language-guided affordance segmentation on 3d object”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 14251–14260 (cit. on p. 83).

[109] Shengyi Qian, Weifeng Chen, Min Bai, et al. “Affordancellm: Grounding affordance from vision language models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 7587–7597 (cit. on p. 83).

[110] Cheng Pan, Kai Junge, and Josie Hughes. “Vision-language-action model and diffusion policy switching enables dexterous control of an anthropomorphic hand”. In: *arXiv preprint arXiv:2410.14022* (2024) (cit. on p. 83).

[111] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, et al. “Grounding dino: Marrying dino with grounded pre-training for open-set object detection”. In: *European Conference on Computer Vision*. Springer. 2024, pp. 38–55 (cit. on p. 85).

[112] Alexander Kirillov, Eric Mintun, Nikhila Ravi, et al. “Segment anything”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2023, pp. 4015–4026 (cit. on p. 86).

[113] Toan Nguyen, Minh Nhat Vu, An Vuong, et al. “Open-vocabulary affordance detection in 3d point clouds”. In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2023, pp. 5692–5698 (cit. on pp. 86, 103).

[114] Alec Radford, Jong Wook Kim, Chris Hallacy, et al. “Learning transferable visual models from natural language supervision”. In: *International conference on machine learning*. PMLR. 2021, pp. 8748–8763 (cit. on p. 86).

[115] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. “Pointnet++: Deep hierarchical feature learning on point sets in a metric space”. In: *Advances in neural information processing systems* 30 (2017) (cit. on p. 88).

[116] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017) (cit. on p. 89).

[117] E. Rohmer, S. P. N. Singh, and M. Freese. “CoppeliaSim (formerly V-REP): a Versatile and Scalable Robot Simulation Framework”. In: *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*. 2013 (cit. on p. 103).

[118] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. “Rlbench: The robot learning benchmark & learning environment”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 3019–3026 (cit. on p. 103).

[119] Fangchen Liu, Kuan Fang, Pieter Abbeel, and Sergey Levine. “Moka: Open-vocabulary robotic manipulation through mark-based visual prompting”. In: *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*. 2024 (cit. on p. 104).

[120] Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. “Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation”. In: *arXiv preprint arXiv:2409.01652* (2024) (cit. on p. 104).

[121] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, et al. “Grounding dino: Marrying dino with grounded pre-training for open-set object detection”. In: *arXiv preprint arXiv:2303.05499* (2023) (cit. on p. 108).

[122] Alexander Kirillov, Eric Mintun, Nikhila Ravi, et al. “Segment Anything”. In: *arXiv:2304.02643* (2023) (cit. on p. 108).