A Digital Twin System for Task-Replanning and Human-Robot Control of Robot Manipulation

Xin Li, Bin He, Zhipeng Wang*, Yanmin Zhou*, Gang Li and Zhongpan Zhu
E-mail address: lixin314@foxmail.com, hebin@tongji.edu.cn, wangzhipeng@tongji.edu.cn, yanmin.zhou@tongji.edu.cn, lig@tongji.edu.cn, 521bergsteiger@tongji.edu.cn
* corresponding authors

College of Electronics and Information Engineering, Tongji University, Shanghai 200120, China
National Key Laboratory of Autonomous Intelligent Unmanned Systems, China
Frontiers Science Center for Intelligent Autonomous Systems, Ministry of Education, China

***Abstract*—In order to enhance the robustness of robots in scenarios with dynamic and complex manipulation tasks and respond quickly to the demands of personalized manufacturing, we propose a DT prototype system named "Alita" to achieve effectively task replanning and human-robot control, inspired by the real-time and closed-loop characteristics of DT. Alita constructs a DT representation with four layers, encoding the geometric, physical and visual dynamics of the work scene, and further obtains unified semantic expressions. Based on this DT representation, Alita establishes two accessible strategies, forming a two-way information feedback loop that endows it with the capability to optimize real robot manipulation. The first strategy presents a deep learning-based model that combines a graph network and a long short-term memory network, and introduces a specialized dataset to replan the manipulation task. The second strategy adopts a multi-virtual force constrained hybrid mapping method of joints and poses to achieve human-robot control. For performance evaluation, we study two cases involving multiple manipulation task replanning and human-robot collaborative grasping. Empirical results demonstrate that Alita accomplishes effectively the tasks of replanning and human-robot control, and mitigates the interferences of environment and task mutations.**

***Index Terms*—Digital twin, robot manipulation, personalized manufacturing, task replanning, human-robot control.**

I. Introduction

Robots have become an important terminal in automatic industrial manufacturing, which is used to complete various manipulation tasks [1]. With the continuous development of robotics technology and the continuous improvement of public living standards, people's demand for personalized manufacturing of small items such as daily necessities and medical supplies is also increasing. The iterative update cycle of various products is significantly shortened, showing the characteristics of dynamic changes in robot manipulation tasks with different operation modes and objects [2]. In addition, in personalized manufacturing scenarios, robot manipulation tasks often have a certain complexity, and in the absence of human intelligence, most robot systems that execute preprogrammed instructions in a conventional manner are difficult to complete these personalized tasks well [3, 4]. Therefore, enabling the system to maintain adaptability to human-robot control while effectively replanning tasks is crucial for improving the operation capabilities of robots in complex dynamic tasks and environments [5]. Currently, there are many studies on the task replanning and control of robot manipulation. The former focuses on learning the relationship between task, scene features and robot actions [6-9] while the latter focuses on using multisource scene information to assist human-robot control [10-13]. Both require the comprehensive perception of the current scene information in time. For example, when a robot performs an assembly task, it needs to obtain timely information such as appearance attributes, positions of objects, robot state, spatial

relationship between objects, and even the physical attributes of objects (which can affect the order in which objects are operated). When faced with the task changes, this information will facilitate quickly replanning of the new task. Moreover, this information can improve operator understanding of the operation process, stimulate their initiative, and provide specific guidance for the robot control. However, the task changes will cause the random addition and removal of objects in the scene, forming an open manipulation scenario. Realizing timely and comprehensive perception of scene state in open operation scenarios with dynamic environments and heterogeneous data is a challenge.

Driven by the concept of the metaverse, digital twin (DT) technology has developed rapidly, providing an innovative way to solve the above problems. DT is an emerging information technology that builds virtual replicas based on models, sensor data and related domain knowledge to accurately map physical space in real time [14-20]. By building a DT representation of robot manipulation scene (RM-DT) using mechanism and data-driven modeling methods, real-time perception of geometric and physical feature changes of scene, and tracking of spatial and temporal dependencies of robot-entities in the scene can be achieved [16, 19, 21]. In addition, combined with semantic techniques, the data interoperability between heterogeneous virtual models can be increased and a more comprehensive representation of scene information is formed [20, 22, 23]. This not only provides rich and available scene knowledge for robot task replanning, but also enables users to understand the real robot operation process in a more easily understandable way. More importantly, semantic DT can integrate robot task replanning and human-robot control into a unified framework [24], endowing the system with capability of multielement decision-making. Especially when RM-DT converges with the real RM system in geometry and physics, it can be used to pre-simulate the replanning results and user control inputs that are fed back to the real robot system. Given these advantages, DT is considered a powerful approach to monitor and optimize robot systems [10, 15-20]. However, differences in research purpose have led to significant differences in the construction of robot-DT system in existing works, especially the lack of consideration for the composition and presentation of DT systems with multielement decision-making functions. Therefore, it is difficult to provide practical reference for deploying DT in replanning and control of robot manipulation. Specifically, some challenges still exist. First, a complex environment hinders the establishment of a feasible DT. Second, based on the rich heterogeneous information extracted by DT, how to realize bidirectional information flow between digital and physical systems to optimize the real robot manipulation.

Motivated by these considerations, we propose a DT prototype system for robotics named "Alita", which offers a semantic-enhanced digital representation of robot work scenes and two accessible interfaces of task replanning and human-robot control. Compared to the existing industrial robot-DT systems, Alita has clear structure, specific implementation approaches, and certain portability. Our preliminary works show that semantic-enhanced robot-DT can quickly track the multi-level changes of the scene [19-20]. This paper focuses on exploring how to improve the robustness of robot under dynamic and complex manipulation tasks based on the semantic-enhanced robot-DT. The main contributions of this paper are threefold: 1) a practical DT system that integrates manipulation task replanning and human-robot control is developed; 2) based on the scene semantics, a manipulation task replanning method using graph neural network (GNN) that is capable of rapidly generating robot action sequences is proposed; 3) a hybrid mapping method of human-robot motion with multi-virtual force constraints is introduced, which facilitates the participation of the operator in the task.

The remainder of this paper is organized as follows: Section II introduces the related works. Section III presents an overview of the framework of the proposed system. The intelligent mapping method of robot manipulation scene is briefly described in Section IV. In Section V and VI, the feedback policies of replanning and control are presented. Subsequently, verification experiments are described in Section VII, and the conclusion is discussed in Section VIII.

II. Related Work

A. Robot Manipulation Task Replanning

In the early stage, robot task planning is often formalized as a deductive or propositional satisfiability problem and combined with symbolic planning, which can be solved by using various search algorithms to obtain feasible action sequences [25]. However, such logical language-based methods use discrete symbols to describe the relationship between environmental states and objects and are unable to describe the continuous information that changes necessary for task planning, such as the spatial position and physical attributes of objects [26]. The two key issues of robot task planning are how to obtain rich scene information and dynamically represent it, and how to integrate this information into robot action sequence generation [27]. In recent years, explosive breakthroughs in artificial intelligence greatly promoted the

research of robot task planning based on learning, providing many methods to solve the above problems. First, with the development of computer vision technology, multidimensional prior knowledge useful for task planning is available, such as the shape, class, attributes of objects [6], spatial relationships between objects [7] and contextual information of objects [28]. However, scene information extracted in these works is limited. As a holistic scene representation, the scene graph stably presents supporting relationships between scene information and is regarded as a potential direction in the research of robot complex task planning [6, 8]. Second, rise of robotic simulation platform accelerates development of deep learning algorithms [29] and various planning strategies of robot action sequence are developed. Most related studies are based on algorithms such as deep reinforcement learning [9, 17] and long short-term memory (LSTM) [30, 31]. The former is suitable for dynamic and partially observable scenarios. However, it is difficult to effectively apply to manipulation tasks with non-trivial causal dependencies and large action spaces, and the *simulation to reality* of strategies is also a challenge [32]. The latter can capture the dependency relationships between sequential actions of robot in different time ranges and it more suitable for rapid robot task planning under limited computing resources [31]. Compared to task planning, task replanning requires faster resolution of action sequences; therefore, it is critical to extract, encode and decode scene information efficiently.

B. Human-Robot Control

Upper-level task planning can mitigate the impact of dynamics of task and environment. However, for complex manipulation tasks like welding [33], it is necessary to introduce humans to the planning layer, which also has great potential for application in the large-scale personalized products [10]. Human-robot control developed earlier and is essentially human robot interaction. Generally, interaction methods include the equipment-based contact type and unmarked visual type [34], among which virtual reality (VR) device-based methods are commonly used because of their better convenience and cost-effective [10, 11, 27]. However, most existing VR-based methods only provide 3D graphics of the scene [11, 33], lacking a description of multi-dimensional information about the scene, such as robot state and object attributes, which makes it difficult for users to intuitively understand the real RM scene and limits the efficiency of human-machine control. How to efficiently map data from human space to robot space is focus of human-robot control. Basic modes of human-robot space mapping involve joint-joint mapping [34] and point-point mapping [10]. The former is intuitive and easy to use, while the latter can ensure good positioning accuracy. Therefore, hybrid mapping methods is considered as effective human-robot control method by combining merits of two modes [35], such as position-rate control and position-single angle control [11, 36]. However, the implementation of these methods is complex and practicality is limited. In addition, due to subjectivity, users may make occasional errors during the operation process, resulting in incorrect actions of the controlled robot. Therefore, the robust human-robot control cannot be guaranteed by these methods solely. In order to constrain the user operations and improve the quality of data mapping, some feedback information is added in many researches, like repulsive force [12] and vision [13]. However, most of these studies only consider a single constraint and cannot be applied to the whole RM space (i.e. free motion and robot-environment contact), and require additional feedback devices [13, 36]. In summary, current scene state and certain prior knowledge of scene are premises to achieve effective task replanning and human-robot control.

C. Digital Twin of Robotics in Manufacturing

DT originates from product life cycle management and has attracted widespread attention in fields such as manufacturing, medicine and Internet of Things driven by the improvements of sensors and modeling technology [37]. Although there are over 20 definitions of DT so far, it is universally accepted that DT system is able to accurately map physical system in real time [14, 38], which makes DT more advantageous in monitoring and optimization of system/process in manufacturing. For industrial robotics, DT can be used for monitoring robot states [15, 20], assembly /disassembly sequence planning [16, 17], grasping learning [39], navigation [40], collision-free path generation [18], human-robot control [10, 33], etc. Most of these works are based on the commonly used three-dimensional or five-dimensional frameworks of DT. The former is composed of *Physical Space*, *Digital Space* and *Connection* with a clear and simple structure [41]. The latter adds two dimensions of *Service* and *Data* to provide a more complete DT [42]. However, due to different research purposes, many works modify the basic framework, such as weakening *Connection* [15] or adding the layer of visualization and user interfaces [10], as well as propose various construction methods of digital space like single geometric modeling [16] and geometric-physical-rule-behavior modeling [17]. In order to enhance the understanding of complex systems, some scholars proposed semantic DT in the manufacturing, exploring how to construct semantic DT based on knowledge graphs [20, 22, 23] or graph representations [37]. Through the semantic DT, not

only can the attributes of entities and relationships between entities in different domains be aggregated [20], but system decisions can also be achieved by deep graph learning [22]. However, these studies are focused on a single decision-making task, such as assembly task rescheduling [23], and lack of research on the composition and representation of DT systems with multielement decision-making functions. In addition, with the development of computer graphics, a number of simulation platforms suitable for robotic vision tasks such as grasping and navigation, have been open-sourced (e.g., iGibson and RoboTHOR [29]), which provide technical guidance for constructing robot-DT system. However, the lack of real-time data links between the digital and physical space makes it difficult for such platforms to be used in dynamic scene.

III. Overview of the Alita

A. System Framework

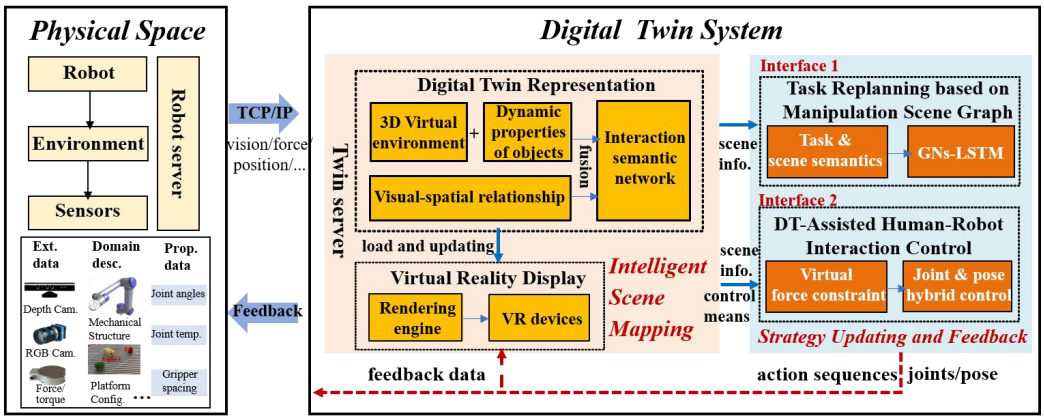

Fig. 1 Overview of the Alita. Best viewed in color.

The framework of proposed system, as illustrated in Fig. 1, comprises of an *Intelligent Scene Mapping* module and a *Strategy Updating and Feedback* module. The former perceives multimodal input from the physical space and maps it into the DT representation. Subsequently, based on current DT representation, the latter attempts to outputs feedback data, which is then sent them back to real robot system, ensuring the smooth completion of the manipulation task. Specifically, the proprioception data of the robotic arm and relevant visual and force sensor data are initially collected through robot server and are transmitted to digital space. Then, based on geometric, physical and relational models, the twin server performs forward iteration to generate three-dimensional virtual environment, physical attributes of objects and visual-spatial relationships between objects. This information is integrated to form a semantic network. Simultaneously, the virtual robot, operable virtual objects embedded with physical attributes, and the semantic network are loaded into the rendering engine and presented in an immersive manner through VR devices. To ensure real-time operation, the 3D graphic models are pre-established and loaded, with only the position and state being updated. Additionally, Alita provides two types of feedback interfaces:

1) when task or scene layout changes, the task replanning interface (interface 1) is activated automatically. The optimized action sequences are generated by task replanning model based on task and current scene information, which is then outputted as feedback data;

2) when the task requires human intelligence, the human-robot interaction control interface (interface 2) is activated manually. User observes the rendered scene and output operation instructions as feedback data through VR devices. It's worth noting that in order to keep track of scene changes, after the task replanning interface is activated, the replanning model is triggered when there are changes of the number of objects or the relationship between objects in the semantic network changes, or the task switching.

B. Problem Setup

We develop a comprehensive formulation to obtain a robust description of Alita. Consider a scene in which a robot R executes manipulation task in an environment deployed with objects $\boldsymbol{E} = \{e_1, ..., e_{|E|}\}$. Let P_d be the data collected in physical space and the mapping function from physical space to digital space is represented by $f(\cdot)$. Then the RM-DT that is denoted by $\boldsymbol{D}_{RM}$ at discrete time point t can be expressed succinctly as Eq. (1). It consists of the virtual replica V_x of physical robot and environment, and the dynamic semantic network Θ that describes manipulation process. When the *Strategy Updating and Feedback* module is activated, with the real-time update of $\boldsymbol{D}_{RM}$, the feedback data ∂ at t can be updated based on currently selected feedback interface and the feedback data at previous time, which can be

written as Eq. (2), where Q_j (j=1, 2) is the decision function of j-type feedback interface. Specifically, ∂ refers to the atomic action sequences that robot needs to perform, or the joint angles and end pose that robot needs to reach.

$$\boldsymbol{D}_{RM,t} = f(P_{d,t}) = [V_x; \Theta]_t \quad (1)$$

$$\partial_t = Q_j(\partial_{t-1}, \boldsymbol{D}_{RM,t-1}) \quad (2)$$

Therefore, in order to achieve effective operation of Alita, the primary objective is to build suitable $f(\cdot)$ and $Q(\cdot)$. To facilitate understanding of this work, some key notations are listed in Table I.

TABLE I List of some key notations

Notation	Description
$(R, \boldsymbol{E})$	Robot and objects set $E = \{e_1, \ldots \ldots, e_{\lvert E \rvert}\}$ in manipulation task
P_d	Data collected in the physical space
$\boldsymbol{D}_{RM}$ /RM-DT	Digital twin representation of robot manipulation. It consists of the simulation-based replica V_x and the dynamic semantic network Θ that describes the manipulation process ($\langle e_i, p_i, value \rangle$ and $\langle e_i, r_{i,j}, e_j \rangle$).
(r, p)	Binary relation semantics (e.g., *relative position*), unary attribute semantics that is divided into class l and other attributes m (e.g., *stiffness, color*)
$(\boldsymbol{I}, \boldsymbol{G}, \boldsymbol{T})$	Scene image, weighted-directed graph reconstructed from Θ and given task
$\boldsymbol{\Lambda}$	$\boldsymbol{\Lambda}_k = \{A_1, A_2, \ldots, A_k\}$ is the atomic action sequences and is set as the feedback data ∂ when task replanning interface is activated. Each atomic action A is expressed as a two-tuple <*operation, object*> (a, e).
$(\boldsymbol{h}_a, \boldsymbol{h}_p)$	Rotation angle and the pose of trackers that are attached to the user's arm
$(\boldsymbol{d}_a, \boldsymbol{d}_p, d_s)$	Joint angle, end pose of the real robotic arm and the gripper spacing. It is set as the feedback data ∂ when human-robot control interface is activated.
F_{fd}	Virtual feedback force generated in digital space. It consists of virtual space force $\bar{F}_d$ and the virtual contact force $\bar{F}_p$

IV. Intelligent Scene Mapping

The *Intelligent Scene Mapping* module provides comprehensive and in-depth understanding of the physical system, and is the foundation for efficiently task replanning and controlling of robot manipulation through Alita. This section briefly introduces the composition of RM-DT and its display method in digital space. Due to limited space, please refer to our preliminary works for detailed information [19, 20].

A. Digital Twin Representation of Robot Manipulation

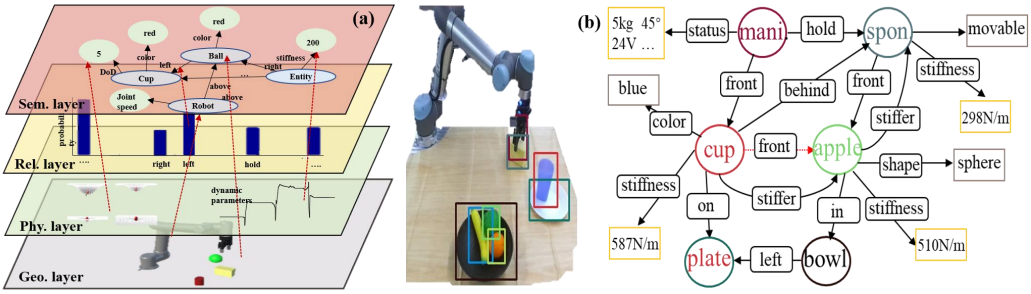

Fig. 2 (a) Four-layer structure of RM-DT. (b) A semantic network instance in RM-DT.

In order to extract multi-level scene information and serve both task replanning and human-robot control, a semantic enhanced RM-DT with four-layers is constructed, which is shown in Fig. 2(a).

- **The first layer** is the geometric layer that consists of 3D graphic models of robot and objects in physical space, providing geometric information like color and shape. The graphic model of robot can be sourced directly from manufacturer's software library. Moreover, with the help of computer vision technology, the accurate graphical models of objects are easily achieved through point cloud-based 3D reconstruction method.
- When the virtual robotic arm is controlled by real data to execute the task, driven by the forward/inverse kinematics and contact dynamics, **the second layer**- physical layer- simulates dynamic interaction behavior and provides estimated physical attributes of objects, including stiffness and damping. To meet real-time operation requirements, the *Kelvin-Voigt Model*, offering clear physical explanation, is adopted as the contact dynamics model [43]. Consequently, the virtual contact force $\widehat{F}_p$ generated in RM-DT is expressed as Eq. (3), where $\widehat{K}$ and $\widehat{B}$ represent the stiffness and damping of real objects that are estimated by the *Self-Perturbing Recursive*

Least-Squares algorithm, $\hat{x}_v$ and $\dot{\hat{x}}_v$ are the penetration depth and velocity of the contact point measured in the first layer, respectively.

$$\widehat{F}_p = \widehat{K}\hat{x}_v + \widehat{B}\dot{\hat{x}}_v \tag{3}$$

- **The third layer**, termed the relational layer, provides the visual relationships between objects in physical space, such as *above*, *with* and *front*. In Alita, a relation recognition model that has fast and accurate recognition capabilities based on visual-spatial-linguistic feature fusion is employed, as proposed in our previous work [20]. The first three layers forms the virtual replica V_x.
- **The fourth layer**, known as the semantic layer, is constructed through dynamically integration and inference of the geometric, physical attributes of objects and the visual relationships between objects in first three layers, and the robot states. It serves to provide comprehensive information about the manipulation scene. Assuming that p and r represent the attribute semantics (e.g., *class*, *color*, *shape*, *stiffness*) and binary relation semantics (e.g., *relative position*, *relative hardness*), respectively, the dynamic semantic network Θ encompasses a series of attribute triplets $\langle e_i, p_i, value\rangle$ and relation triplets $\langle e_i, r_{i,j}, e_j\rangle$, such as $\langle e_i, stiffness, 200N/m\rangle$ and $\langle e_i, left, e_j\rangle$. Fig. 2 (b) illustrates an example of the semantic network in the RM-DT.

B. *Virtual Reality Display*

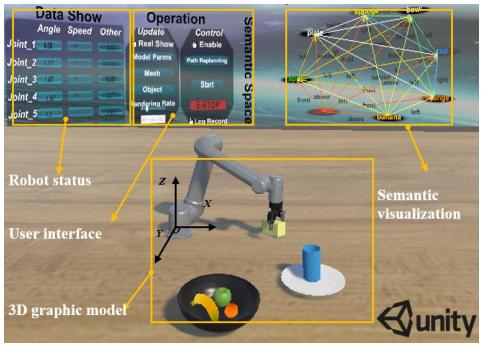

Fig. 3 Virtual reality display of RM-DT.

In practice, Unity 3D and the VR device HTC VIVE are integrated to create an interactive and immersive digital space that displays RM-DT (as shown in Fig. 3). In the digital space, all 3D objects acquire physical behavior through scripts and are governed by the integrated physics engine PhysX. Beyond fundamental functionalities like collision detection and data/scene browsing (including robot operation data, contact force, payload and semantic network), the digital space also incorporates functional extensions such as emergency stops, modification of updating rate, generation and updating of model parameter, and log savings.

V. Task Replanning Based on RM Scene Graph

Faced with open operation scenario, the action sequences that robots need to execute may appear in different combinations. For example, when executing the task "put object A into object B", a new task "put object B on the table" is received, and the action sequence needs to be replanned to complete the new task, such as "place the object A "-> "grasp the object B". In order to encode the dynamic scene effectively and solve the robot action sequences quickly required to complete the specified task, this section propose a GNs-LSTM-based manipulation task replanning model, benefiting from the rich and real-time scene knowledge provided by RM-DT, and a specialized dataset including desktop manipulation tasks with different granularity is compiled for model training and verification.

A. *Description of Manipulation Task Replanning*

The key to task replanning is to decompose the task into multiple atomic actions, which cannot be segmented. To reduce the solution space dimensions, in Alita, the atomic action that is expressed as a two-tuple <*operation, object*> refers to the basic operation of the robotic arm on the associated object, like <*approach*, *kettle*>. Taking the “get a cup of water” as an example, the task decomposition process can be described by the AND-OR diagram, which clearly represents the hierarchical structure of various atomic actions in task (as shown in Fig. 4(a)). In a certain scene, the corresponding atomic action sequence is represented by Fig. 4(b). Notably, the result of task replanning is related to the task itself, the completed atomic actions and the current scene information.

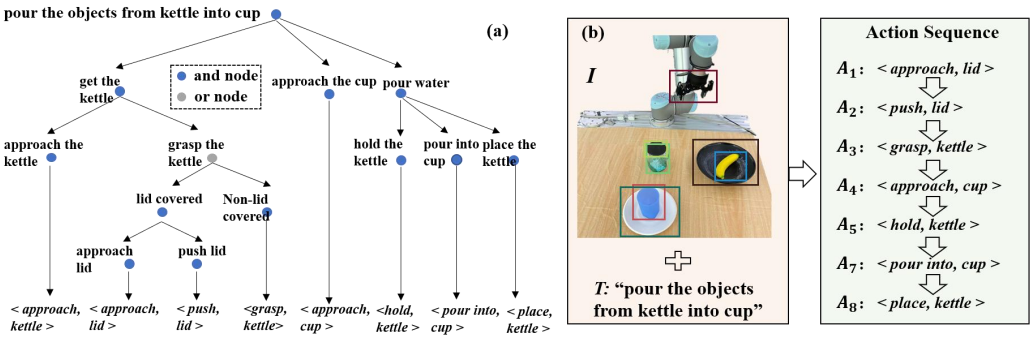

Fig. 4 (a) Decomposition process of "*get a cup of water*" using AND-OR diagram. (b) Atomic action sequence in a certain scene.

Based on the above analysis, the task replanning problem can be expressed in the form of probability estimation. Let $\boldsymbol{I}$ and $\boldsymbol{T}$ be the current scene image and the given task, respectively, and $A_k = (a_k, e_k)|a_k \in \boldsymbol{O}, e_k \in \boldsymbol{E}$ be the atomic action at step k, where $\boldsymbol{O}$ represents the set of basic operations. Based on the chain rule, the probability of atomic action sequences for a given task can be recursively decomposed into Eq. (4), where $\boldsymbol{\Lambda}_{k-1} = \{A_1, A_2, ..., A_{k-1}\}$. Assuming that the distribution between the basic operation and associated object are independent and can be predicted separately, the probability of atomic action at k can be obtained according to Eq. (5).

$$p(A_1, A_2, ..., A_n|\boldsymbol{I}, T) = \prod_{k=1}^{n} p(A_k|\boldsymbol{I}, T, \boldsymbol{\Lambda}_{k-1}) \tag{4}$$

$$p(A_k|\boldsymbol{I}, \boldsymbol{T}, \boldsymbol{\Lambda}_{k-1}) = p(a_k|\boldsymbol{I}, \boldsymbol{T}, \boldsymbol{\Lambda}_{k-1}) p(e_k|\boldsymbol{I}, \boldsymbol{T}, \boldsymbol{\Lambda}_{k-1}) \tag{5}$$

As mentioned earlier, the semantic network represents the geometric and physical attributes of objects and the relationships between objects, and can provide rich scene prior knowledge for task replanning. In order to facilitate the extraction of attribute features of objects and structure features of scene, the semantic network Θ is reconstructed into a weighted-directed graph $\boldsymbol{G}$=($\boldsymbol{V}$, $\boldsymbol{S}$) instead of scene image to predict the atomic action sequence, where $\boldsymbol{V} = \{(e_i, p_i)\}, i = 1, ...|\boldsymbol{E}|$ is the node set of a graph, $\boldsymbol{S} = \{(e_{j,s}, e_{j,e}, r_j)\}, j = 1, ...|\boldsymbol{S}|$ is the edge set of a graph, the $e_{j,s}$, $e_{j,e}$, r_j are the sender node, receiver node, and edge attribute of the j-th edge, respectively. Therefore, Eq. (5) can be rewritten as Eq. (6).

$$p(A_k|\boldsymbol{G}, \boldsymbol{T}, \boldsymbol{\Lambda}_{k-1}) = p(a_k|\boldsymbol{G}, \boldsymbol{T}, \boldsymbol{\Lambda}_{k-1}) p(e_k|\boldsymbol{G}, \boldsymbol{T}, \boldsymbol{\Lambda}_{k-1}) \tag{6}$$

B. Action Sequence Prediction based on GNs-LSTM

Given the powerful capabilities of GNN [22] and LSTM [30] in complex relationship modeling and sequence data processing, respectively, this section combines the two networks to simulate the process of atomic action sequence prediction, i.e. Eq. (6). The proposed model is illustrated in Fig. 5. First, $\boldsymbol{G}$ and $\boldsymbol{T}$ are processed based on the GNs and *word embedding* to obtain the graph embedding vector $\boldsymbol{f_G}$ and task embedding vector $\boldsymbol{f_T}$, respectively. Then, these two vectors are concatenated and input to the LSTM for decoding to obtain the atomic actions at each step.

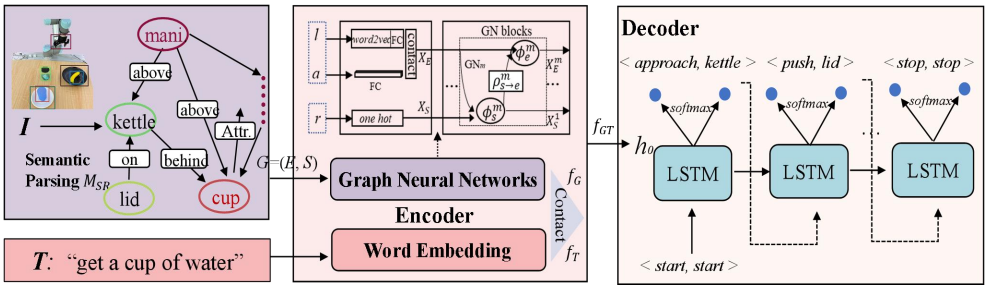

Fig. 5 Prediction model of atomic action sequence based on GNs-LSTM.

***1) GNs-based Feature Encoder*:** Compared with other deep learning-based analysis methods of graph domain information, such as GCN and GAN, the GN has a strong relation induction ability and is suitable for processing weight-directed graph [44]. Therefore, it is adopted to learn the node representation in our network. First, the class l and other attributes a (e.g., *stiffness*) of the node in the scene graph are, respectively, dimensional increased through *Word2Vec* and the FC layer (the activation function is *softmax*) to form a node feature matrix $\boldsymbol{X}_E = \{X_{e,i}\} \in R^{|E| \times D}$, where D is the dimension of node features. Second, *One-Hot Encoding* is used to represent the edge between the nodes as feature vector $\boldsymbol{X}_S = \{X_{r,j}\} \in$

$R^{|S|\times N}$, where N is the number of edge types. Finally, both are simultaneously input into multi-GN for node iterative updating, and the node feature vectors $\boldsymbol{X}_E^M$ and edge feature vectors $\boldsymbol{X}_S^M$ are obtained as shown in Eq. (7), where M is the number of GN.

$$\left(\boldsymbol{X}_E^M, \boldsymbol{X}_S^M\right) = GN_M\left(GN_{M-1}\left(\ldots GN_1(\boldsymbol{X}_E, \boldsymbol{X}_S)\right)\right) \tag{7}$$

Therefore, the $\boldsymbol{X}_E^M$ that aggregates the information of all node attributes and the graph structure can be used to represent the feature $\boldsymbol{f}_G$, and the input vector of the decoder can be represented by Eq. (8), where $\boldsymbol{W}_{fG}$ and $\boldsymbol{W}_{fT}$ are learnable parameters. To ensure that each scene graph has a feature vector of the same size, $\boldsymbol{f}_G$ is zero-padded according to the maximum number of nodes ($R^{|E|_{max}\times D}$).

$$\boldsymbol{f}_{GT} = \left[relu\left(\boldsymbol{W}_{fG}\boldsymbol{f}_G\right), relu\left(\boldsymbol{W}_{fT}\boldsymbol{f}_T\right)\right] \tag{8}$$

2) LSTM-based feature Decoder: To generate temporal related atomic actions, the LSTM is used to decode scene graph features. At the initial step, $\boldsymbol{f}_{GT}$, the zero vector of the same dimension and the special atomic action <a_0, e_0> = <*start*, *start*> are used to initialize the hidden layer $\boldsymbol{h}_0$, cell state $\boldsymbol{c}_0$ and input of the LSTM cell, respectively. At step k, the hidden state $\boldsymbol{h}_k$ and cell state $\boldsymbol{c}_k$ are first calculated by the atomic action feature $\boldsymbol{f}_k^A$, and then the probability distributions ($p(a_k)$ and $p(\mathrm{e}_k)$) of all basic operations and associated objects are obtained by passing $\boldsymbol{h}_k$ through two FC layers (the activation function is *softmax*). The process of the LSTM-based feature decoder is described as Eq. (9). Similarly, $\boldsymbol{W}_a$ and $\boldsymbol{W}_e$ are the learnable parameters, and b_a and b_e are the corresponding deviation vectors. Among them, the atomic action with the greatest probability is selected as the input of the next sequence until the special atomic action<a_0, e_0> = <*stop, stop*> outputs. It is worth noting that $\boldsymbol{f}_k^A$ is concatenated by the one-hot vectors of the basic operation and associated object at step k-1.

$$\begin{aligned}
[\boldsymbol{h}_k, \boldsymbol{c}_k] &= LSTM\left(\boldsymbol{f}_k^A, \boldsymbol{h}_{k-1}, \boldsymbol{c}_{k-1}\right) \\
p(a_k) &= softmax(\boldsymbol{W}_a\boldsymbol{h}_k + b_a) \\
p(e_k) &= softmax(\boldsymbol{W}_e\boldsymbol{h}_k + b_e)
\end{aligned} \tag{9}$$

C. Dataset Construction

TABLE II Description of Task Replanning Dataset

Item	Description
Overall	Nearly 800 RGB-D scene images, 1,062 sequence samples (G,T), 4,615 atomic actions
Annotations	class, stiffness and color, etc. of the objects, the visual relations between objects, atomic action sequence
Task	**Sorting**— task#1: "organize the desktop", task#4: "put banana onto plate"; **Pour**: task#2: "get a cup of water", task#5: "pour water from cup into bowl"; **Wash**—task#3: "wash the bowl", task#6: "wash the green fruits"
Operations	*approach, grasp, move together, place, pour into, push, stir, hold*
Objects	*sponge, cup, bowl, plate, banana, apple, orange, kettle, lid*

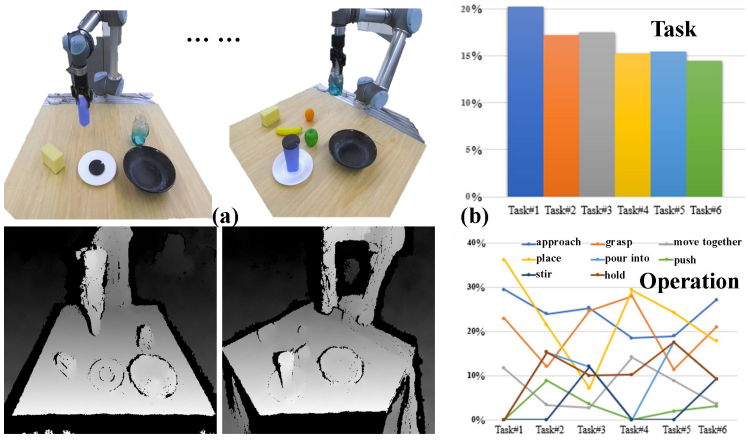

Fig. 6 (a) Data collection scenario. (b) Statistical distribution of tasks and basic operations in the dataset.

To enhance the model's comprehension of the task planning problem, a lightweight dataset of RM task replanning, is constructed, as illustrated in Table II. This specialized dataset contains three distinct types of manipulation tasks related to ***sorting***, ***pour*** and ***wash***, each comprising two subtasks with varying granularity. Fig. 6 (a) shows the RGB-D data collected from different views, while Fig. 6 (b) offers a statistical distribution of task categories and basic operation categories in the dataset. It can be found that the balance of task categories is maintained with minor deviations, ensuring that different tasks get the equal amounts of training. Moreover, it is evident that there is a dependency between basic operations and tasks, which is similar to the actual situation. For example, in task#4: "put banana onto plate", there is no operation "pour", and similarly, there should be stronger dependency between tasks and associated object. More detailed introduction is available in the link in the conclusion.

VI. RM-DT-Assisted Human-Robot Interaction Control

In order to encourage operators to participate in RM and improves the adaptability of robots in complex and uncertain manipulation tasks, this section introduces an RM-DT assisted human robot control method, as shown in Fig. 7. In comparison to existing human-robot control methods [10, 11, 33, 34, 36], the proposed method not only intuitively display scene changes to operators in the form of 3D graphics and semantics but also provides a hybrid human-robot motion mapping method with multi-virtual force constraints for operators.

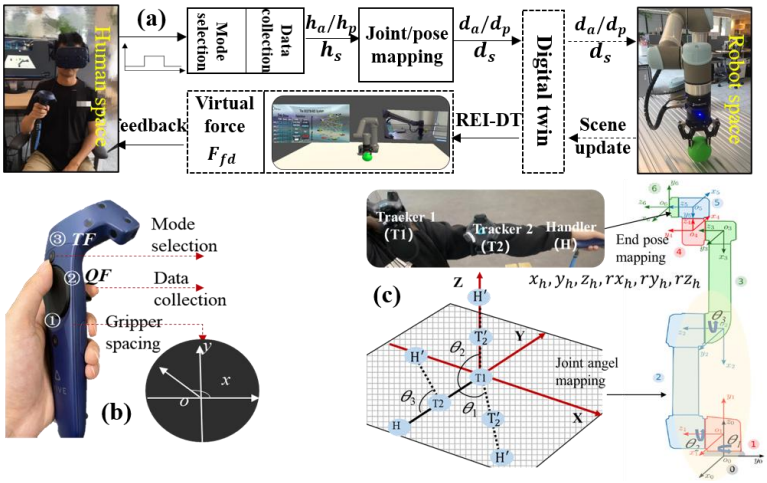

Fig. 7 Proposed method of human- robot control. (a) Overall framework. (b) Function mapping of handle. (c) Mapping method of end pose and joints angle.

Taking VR device HTC VIVE and robotic arm UR5 as examples. First, the operator observes the digital space through the helmet-mounted display and then selects the data mapping mode through Boolean input *TF* of handle. Specifically, the large-scale motion of the robotic arm is controlled by joints angle mapping, and after roughly positioning, the end pose mapping is employed to realize the small-scale motion of the robotic arm. Subsequently, when the Boolean input *QF* of handle is set to 1, the locator records the rotation angle $\boldsymbol{h}_a$ or the pose $\boldsymbol{h}_p$ of trackers attached to the operator's arm, respectively. These values are then sent to digital and robot space to control the robot after being converted into the joint angle $\boldsymbol{d}_a$ or end pose $\boldsymbol{d}_p$ of the real robotic arm. Note that with the data collection function, the operator can adjust to the appropriate posture. When the robotic arm reaches the target pose, the operator provides the gripper distance d_s through handle to open the gripper and grasp the object. In the whole process of robot manipulation, the digital space provides the virtual force F_{fd} as the feedback information to constrain the movement of the operator's arm. In practice, the virtual force is dynamically displayed to the user in the form of a curve, and the user observes the changes of the virtual force in real time during the control process.

A. Data mapping

The mapping method of end pose and joints angle is shown in Fig. 7(c). Considering that in the actual operation process, operator progressively manipulate VR devices based on visual iterations to achieve the target pose, therefore, the incremental accumulation of arm motion data is utilized as the actual control information. **For joints angle mapping**, only the first three joints of robot are considered for two main reasons. First, the large-scale motion of a robot is primarily associated with its first three joints. Second, the heterogeneity between the human arm and robotic arm increases the burden on the operator when

controlling a multi-joint robot. In human space, with tracker1 as the origin, the difference in the rotation angle of the vector composed of handle and tracker1 around the Z-axis, the difference in rotation angle of the vector composed of handle and tracker1 around the Y-axis, and the difference in rotation angle of the vector composed of handle and tracker2 around the Y-axis are used as the motion data of operator's arm, as $\Delta \boldsymbol{h}_a = \boldsymbol{mean}[\Delta\theta_1, \Delta\theta_2, \Delta\theta_3]^{\boldsymbol{T}}$. Here the ***mean*** is used for average filtering on the data to compensate for the influence of arm shaking. Similarly, **for end pose mapping**, the motion data of operator's arm is represented by the difference between the final pose and the initial pose of the handle, as $\Delta \boldsymbol{h_p} = \boldsymbol{mean}[\Delta x_h, \Delta y_h, \Delta z_h, \Delta rx_h, \Delta ry_h, \Delta rz_h]^{\boldsymbol{T}}$.

Let $\boldsymbol{d}_a^o$ be the initial joints angle of the real robotic arm, and $\boldsymbol{d}_p^o$ be the initial pose of the robotic arm end. Then, the operation data of the robotic arm can be represented by Eq. (10), where $\boldsymbol{T}(D, H)$ represents the transformation matrix of coordinate system from human space to digital space, and *TF* represents the data mapping mode, which is a Boolean input.

$$\begin{cases} \boldsymbol{d_a} = [\boldsymbol{d}_a^o + \Delta \boldsymbol{h}_a] & TF = 0 \\ \boldsymbol{d_p} = [\boldsymbol{d}_p^o + \boldsymbol{T}(H, D)\Delta \boldsymbol{h_p}] & TF = 1 \end{cases} \tag{10}$$

For gripper control, the angle between the position vector (x, y) of the finger pressed on the touchpad of handle and the (1, 0) vector is mapped as the gripper spacing. Therefore, the desired spacing d_s can be calculated by Eq. (11), where $S_{gripper}$ is the maximum spacing of gripper.

$$d_s = \left(\text{atan2}\left[\frac{(x,y)\cdot(0,1)}{(x,y)\times(0,1)}\right] + \pi\right) \Big/ 2\pi * S_{gripper} \tag{11}$$

B. Virtual Force Calculation

To improve the operation efficiency, the digital space force $\widehat{F}_d$ and the virtual contact force $\widehat{F}_p$ are employed to constrain the operator's operations. When the virtual force continues to increase and exceeds the threshold, the user adjusts the motion speed and direction or gripper spacing to reduce the virtual force.

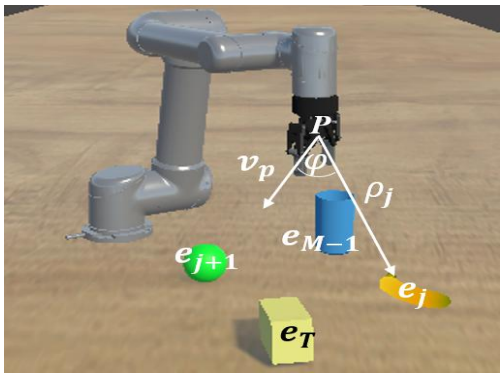

Fig. 8 Gripper and target objects in digital space.

In free space: In each set of operations, the target object is uniquely determined according to the known operation order, and non-target objects can be considered "obstacles". Therefore, when the robotic arm approaches or even touches the non-target objects due to the human operation, it can be considered to be the wrong operation. Based on this, the "Gripper-nontarget Objects" in RM-DT is equivalent to the "Source-Obstacles" in the danger field (DF) [45], and its real-time DF value can be adopted to guide operator to stay away from non-target objects during operation. As shown in Fig. 8, let e_T and $\boldsymbol{E}_b = \{e_1, ...e_j, ..., e_{M-1}\}$ be the target object and the set of nontarget objects in the current manipulation task. v_p and ρ_j represent the movement velocity of the gripper, and the distance between the gripper and e_j, respectively. The DF value between e_j and the gripper can be calculated by Eq. (12) [45], where $k_1 > k_2\|v_p\|_{max}^{\lambda}$. In Alita, the average of DF values of all nontarget objects is defined as the risk coefficient DF_{DT} of the RM-DT, as shown in Eq. (13), where the L_p represents L_p-norm.

$$DF\big(e_j(k)\big) = k_1 + k_2\big\|v_p(k)\big\|^{\lambda} \cos\varphi \Big/ \rho_j(k) \tag{12}$$

$$DF_{DT}(k) = \left(\frac{1}{M-1}\right) L_p\big(\big\|DF\big(e_j(k)\big)\big\| .. \big\|DF\big(e_{M-1}(k)\big)\big\|\big) \tag{13}$$

In order to solve the problem of unreachable motion caused by the close distance between the target object and nontarget object, the variable factor ρ_T, the ratio between the distance from the gripper to the target object and the distance from the gripper to the nontarget objects—is considered. Therefore, the $\widehat{F}_d$ is

written as $\widehat{F}_d = \rho_T DF_{DT}$. For operator, when the $\widehat{F}_d$ continues to increase and exceeds the threshold, they should adjust the manipulation to reduce it.

In contact space: When the robotic arm is in contact with an object, the operator can control the contact force by adjusting the gripper spacing based on $\widehat{F}_p$, which is expressed as Eq. (3).

Therefore, the real-time virtual feedback force F_{fd} can be calculated as Eq. (14).

$$F_{fd} = \begin{cases} \widehat{F}_p & \widehat{F}_p \neq 0 \text{ and } \hat{F}_p \neq 0 \\ \widehat{F}_d & else \end{cases} \tag{14}$$

VII. Case Study

To show the effectiveness of Alita on task replanning and control of RM, we establish a lab-scale platform and conduct two corresponding cases. The customized platform, depicted as Fig. 9, comprises a 6-DOF UR-5 robotic arm with a gripper of Robotiq2F-85, a 6-DOF force sensor of Robotiq FT300, and a ZED camera set aside. The robot server and twin server are run with a 11 GB-GTX1080Ti graphics card at 3.7 GHz and a 4 GB-GTX1050ti graphics card at 2.2 GHz. In addition, the TCP/IP protocol is used to complete data exchange between servers.

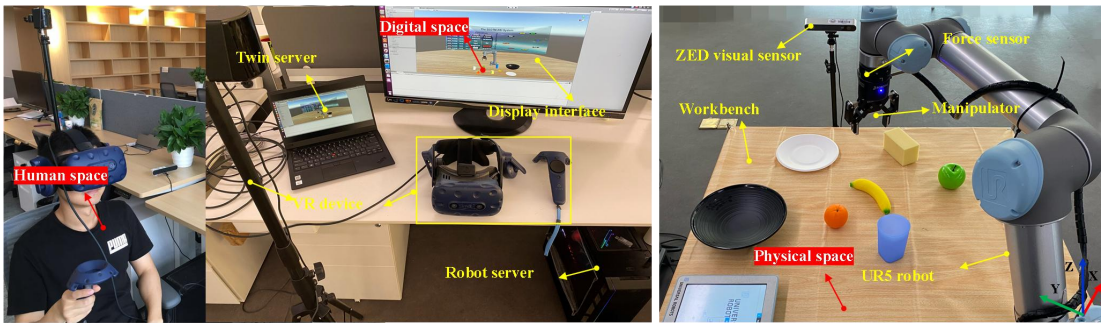

Fig. 9 Setup of real customized platform.

A. Task Replanning Performance

Model Validation and Analysis: The proposed task replanning network is trained using PyTorch on Ubuntu 16.04 with a 32 GB Tesla V100 GPU. The training process employs an RMSProp optimizer with a learning rate = $1e$-3 and decay rate = $1e$-4 are used. The batch size, numbers of epochs, GN and LSTM are set to 30, 200, 2 and 15 (maximum task sequence length), respectively. Four baselines are implemented with the same training parameters as GNs-LSTM: Nearest Neighbor (NN), MLP, MLP-LSTM and GNs-MLP. Moreover, we separately trained the inference network of atomic action sequence without class (Ours- no classM), attribute (Ours- no attr.M) and relation (Ours- no rel.M) input to assess the influence of these three features on the results.

Table III shows the sequence prediction accuracies of all models on task#1~task#6. Overall, the proposed model exhibits a quick prediction of atomic action sequences for different task with an average accuracy of 93.5%, significantly outperforming baseline models in most tasks (5/6) with an accuracy improvement of over 9.2%. Furthermore, we observed the following: 1) in the sequence prediction of manipulation task with strong temporality, the LSTM demonstrates a proficiency in memorizing long-term dependencies between actions; 2) If the GNs are removed, achieving better prediction performance becomes challenging due to the lack of aggregation of edge and node information; 3) different types of tasks have certain deviations in the dependence of features. For instance, in task#4 and task#5, which involve specific operation objects and sequences, the significance of object class and the relation between objects is much greater than that of object attributes; 4) there are disparities in the prediction performance of GNs-LSTM across different tasks. In ***wash***-related tasks such as task#3 and task#6, the detection effect is superior, which may be because of the relatively fixed atomic action sequence; 5) the multi-level scene information provided by RM-DT effectively enhances the prediction performance of atomic action sequences, and the effect of class and relation features is greater than that of attribute feature; 6) the LSTM-based decoder experiences a longer inference time, which is caused by the serial output of atomic actions.

TABLE III Comparison of prediction accuracies of atomic action sequences. The atomic action sequence is considered to be correct prediction if the predicted atomic actions (both the basic operation and associated object) at each time step are correct.

Methods	Accuracy							Average inference time (ms)
	T#1	T#2	T#3	T#4	T#5	T#6	Overall	
NN	31.6	33.5	28.3	30.9	33.9	37.1	32.3	35
MLP	58.6	71.8	62.6	69.8	76.6	63.3	66.2	54
MLP-LSTM	82.7	84.9	85.9	75.7	84.7	88.5	84.3	65
GNs-MLP	73.7	81.7	79.8	71.1	80.1	82.6	78.6	58
Ours	90.5	93.8	95.1	91.8	93.5	95.7	93.5	70
Ours- no classM	85.0	84.7	86.8	76.6	82.1	83.6	82.9	
Ours- no rel.M	83.1	83.7	83.9	73.2	85.0	86.4	83.2	
Ours- no attr.M	83.4	89.4	81.1	93.5	92.8	85.6	86.7	

Fig. 10 summarizes the prediction performance of all methods on basic operations and associated objects. It is evident that, except for NN, most methods can achieve good accuracy on single *operations* or *objects*. However, the overall detection rate of *objects* is lower than that of other methods in the absence of object class. By comparing the prediction results, there is no dependent relationship between *operations* and *objects*, which is consistent with the aforementioned analysis. Moreover, the confusion matrices of the predicted results reveals that the proposed model achieves high accuracy in most class predictions. Nevertheless, there is a certain probability of confusion in operation predictions with strong sequential correlation, such as *approach*, *grasp*, *move together*, and *place*. Similarly, closely related object may also be mistakenly identified, such as *lid* and *kettle* or *banana* and *apple*.

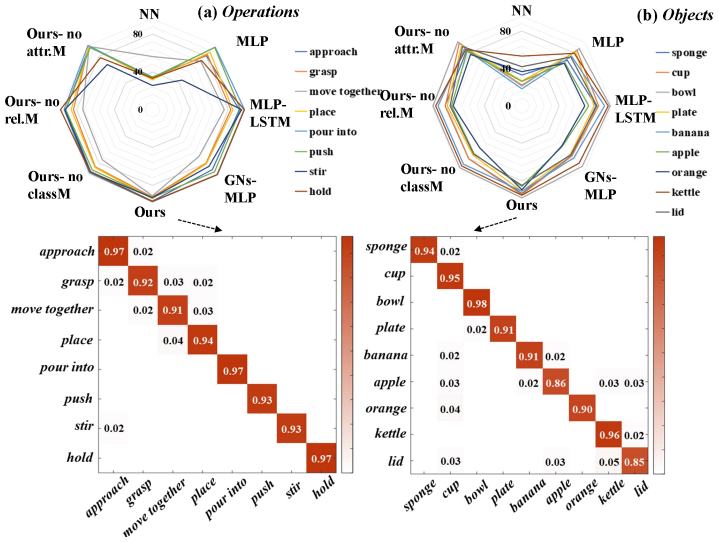

Fig. 10 Comparison of prediction accuracies of (a) operations and (b) objects.

TABLE IV Prediction accuracies on previously unseen tasks.

Methods	T#7	T#8
Ours	85.5	88.3
Ours- no classM	78.9	81.2
Ours- no rel.M	77.6	84.9
Ours- no attr.M	83.7	85.5

To assess the generalization ability of the proposed model, we define two new tasks that have same atomic action or similar time context information with tasks in the dataset: task#7- "put orange into bowl" and task# 8- "point water into kettle". Each task comprises approximately 100 samples (G, T). The test results of the trained model on task# 7 and task# 8 are presented in Table IV. Despite the trained model exhibiting lower prediction accuracy for atomic action sequences on previously unseen tasks compared to those seen before, it still achieves over 85%. This suggests that the proposed method has a certain level of generalization ability for similar tasks.

Multitasking Experimental Verification: Based on the experimental platform, a multitasking experiment is designed to verify the effectiveness of the proposed model in real RM scene. In the workspace, illustrated in Fig. 11(a), the atomic action sequences in the order of pre-set tasks are generated by the trained GNs-LSTM and transmitted to the robot server to drive the real robot motion. During the manipulation process, 6 types of interference are set as shown in Fig. 11(b). For implementation method of attaching the atomic actions to motion of real robot, see the APPENDIX. B.

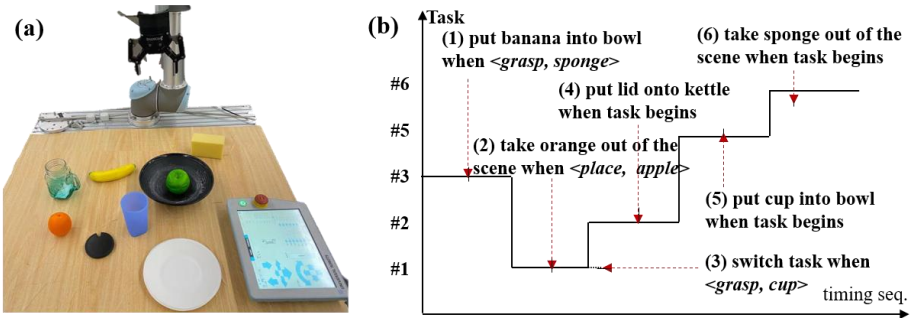

Fig. 11 Multitasking experiment (a) workspace, (b) task setting.

Fig. 12 records the atomic action sequences ultimately executed by the robotic arm during the entire task. Compared with the original task setting, when confronted with sudden environmental changes, the proposed method can rapidly plan the action sequences based on the current scene states and mitigate the impact of environmental changes on the manipulation task. For instance, in (1), a sequence of atomic actions <*grasp*, *banana*>, …, <*place*, *banana*> is generated before <*stir, sponge*> to clear the obstacle <*banana*, *in*, *bowl*>. Furthermore, in cases of sudden task switching, the proposed method can also correctly replan the action sequences according to the new task, which eliminates the confusion in action caused by task mutation. For example, after suddenly switching from task#1 to task#2, the <*place*, *cup*> can be generated at the initial time, ensuring the smooth progression of the current task. This result verifies the effectiveness of the proposed task replanning method based on DT, providing reliable data support for the feedback of action sequence from digital space to physical space.

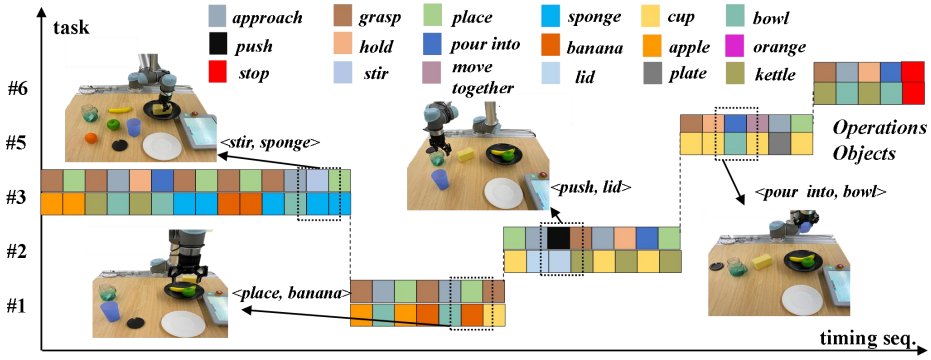

Fig. 12 Result of task replanning.

B. Human-Robot Collaborative Grasping

Experiment Setting In order to verify the effect of RM-DT-assisted human robot control method. This section implements two experiments of human-robot collaborative grasping, which is shown in Fig. 13. In the experiment#1- "*Target object grab-and-place*", an operator controls the robotic arm from the starting point ***S*** to complete the *approach*, *grasp* target object **A**, and move to end point ***T*** and place object **A**. In the experiment#2- "*Multiple obstacles avoidance*", we take object **B** as the target object and place obstacles **A** and **C** near **B**, and operator controls the robotic arm from the ***S*** to complete the *approach* and *grasp* **B**. We recruited four students with robotics backgrounds (two males and two females) to control the UR5 robotic arm to perform these two tasks separately. Before the experiment, users know the experiment task and be familiar with the use method of VR devices. The parameters in Eq. (12) and (13) are, respectively, $k_1 = 2, k_2 = 0.25, \lambda = 1$ and p=1 [19]. Moreover, the thresholds for $\widehat{F}_d$ and $\widehat{F}_p$ are set to 10 and 20, respectively.

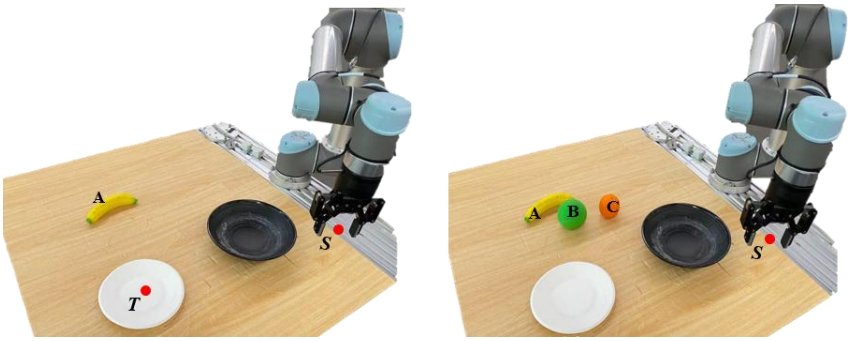

Fig. 13 Human-robot cooperative grasping experiment. (a) Target object grab-and-place. (b) Multiple obstacles avoidance.

Results Table V records the number of user operations required to complete the task and the position tracking errors between manipulator and user's hand, with or without hybrid mapping method, digital space and virtual force feedback. Considering that the mean absolute error (MAE) performs more robustly in the presence of outliers (which may occur during user control), therefore, the MAE is used as a representation of error in the experiment. It can be seen from the results that in experiment#1 and #2, the average number of user operations are 9 and 8 based on our proposed method, which are reduced by at least 5 and 2 times compared to the that of baseline method, respectively. Moreover, the MAE of position tracking does not exceed 1mm (as averaged from the experimental results of 4 students), indicating good accuracy of human-robot control. It is worth noting that, although both hybrid mapping and digital space can improve the efficiency of human-robot control, hybrid mapping increases the MAE of position tracking (from 0.6mm to 0.9mm). This may be due to flaws in the method of calculating user joint angles based on trackers attached to the joints and handheld handler. Thanks to the RM-DT with 3D graphics and semantic representation, our proposed human-robot control method can be easily implemented.

TABLE V Comparisons of human-robot control effect under different RM-DT auxiliary forms. - means unavailable. By default, without hybrid mapping and digital space, the user controls the robot using pose mapping and observes the real RM scene through video.

Experiment item	*Target Object Grab-and-Place*			*Obstacles Avoidance*	
	i	*ii*	*iii*	*i*	*ii*
Hybrid mapping	-	-	√	√	√
Digital space	-	√	√	√	√
Virtual force	-	-	-	-	√
Number of user operation ↓	21 ±2	14 ±2	9 ±2	10 ±1	8 ±1
Position tracking MAE (mm) ↓	1.0± 0.20	0.6± 0.12	0.9± 0.17	0.8± 0.15	0.6± 0.10

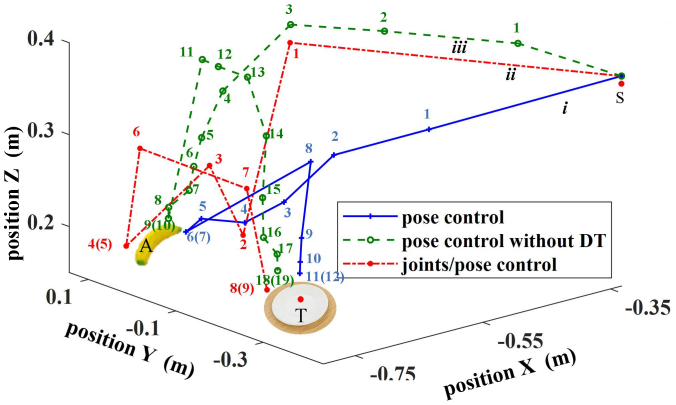

Fig. 14 Manipulator motion trajectories of the robotic arm in robot space. The numbers in parentheses indicate *grasp* or *place*.

In order to show more concretely the effectiveness of the proposed method, a manipulator motion trajectory in the robot space during an experiment#1 is recorded, which is shown in Fig. 14. With the joint/pose hybrid mapping, the task can be completed with a better motion path and has higher manipulation efficiency than pose control because it can not only cover the motion space of the robotic arm, but also perform fine operations. Moreover, compared to video observation, observing real RM scene based on digital space greatly reduces the number of user operations. This is because the former may not provide intuitive spatial information of scene, and more explorations are required, especially when **A** is about to be touched or ***T*** is reached (before and after the 9th, 18th in curve *iii*).

Similarly, a manipulator motion trajectory in the robot space during an experiment#2 is shown in Fig. 15. The results show that through the continuous iteration of the motion-virtual force feedback process, users can control the robotic arm to reach the target position with fewer operations and with a relatively optimal motion path (curve *ii*), which is closer to the optimal path represented by the green line (the shortest path

and no collision). In addition, comparing the danger field values of obstacles **A** and **C** with the system, we can see that the DF value of the system with virtual feedback force is smaller than that without virtual feedback force. This also indicates that virtual force feedback provides a constraint to reduce user operational errors in some extent. Specifically, it can be observed that due to the proximity of target object **B** to obstacle **A**, without virtual force feedback (curve *i*), the operator mistakenly controlled the robotic arm to approach **A** (4th) after controlling it to reach above B (3th), resulting in a rapid increase in the system's DF value. When there is virtual force feedback (curve *ii*), the operator can correctly adjust the motion direction to approach the **B** (4th) after controlling the robotic arm to reach above **B** (3th). Obviously, the virtual force provides an operational constraint that helps the operator, to some extent, choose a better path when there is no force feedback device.

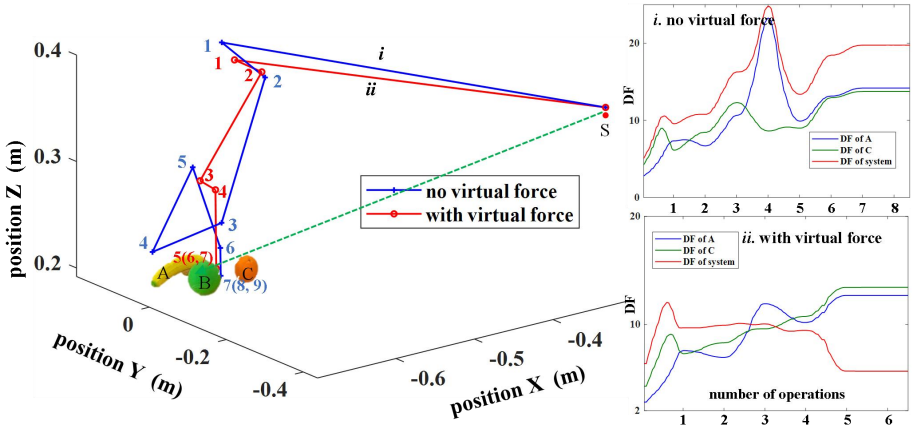

Fig. 15 Manipulator motion trajectories of robot with/without virtual feedback.

VIII. Conclusion and Discussion

A. Conclusion

In this study, we proposed Alita, a novel DT prototype system for manipulation task replanning and human-robot control, which provides guidance for robustly deploying robots in scenarios with dynamic and complex manipulation tasks. The Alita comprises of a DT representation with four-layer that expresses dynamically geometric, physical and relation information of the scene, a GNs-LSTM-based manipulation task replanning interface along with a virtual force-constrained joints/pose hybrid human-robot control interface. To validate the effectiveness of Alita, we compiled an action sequence dataset and designed two experiment cases. The experimental results prove that Alita facilitates avoidance of potential interferences for robots: 1) the task replanning model efficiently generates action sequences of tasks with an average accuracy of 93.5% and inference time of 70ms. The accuracy is at least 9.2% higher than baseline method (6.8% higher than situations with only single information); 2) the number of user operations is reduced by approximately 5 times, providing robots with motion path with fewer collisions. In future work, to enhance Alita's robustness, we aim to study the self-correction and evolution mechanism of the DT model by incorporating the intelligent algorithms such as RL. In addition, our focus will extend to promoting Alita in more challenging physical human-robot collaboration scenarios.

B. Discussion

With the continuous improvement of people's living standards and aesthetics, there is a growing demand for personalized manufacturing of small items, including daily necessities and medical supplies. Addressing the rising need for personalized goods, enhancing flexible production capacity stands out as a key breakthrough for the sustainable development of manufacturing enterprises. This paper proposed a method for effectively manipulation task replanning and human-robot control based on digital twin. A dataset of robot personalized manipulation tasks related to ***sorting***, ***wash***, and ***pour*** was established for planning method validation. The effectiveness of the human-robot control method was verified through two types of tasks: *target object grab-and-place* and *multiple objects avoidance*.

In addition to its main innovative points, Alita has a clear and lightweight structure and is scalable due to the independent nature of two feedback interfaces. Developing new functional interfaces will not affect existing interfaces. For instance, the 3D graphic model of the scene and various spatial information provided by RM-DT can be utilized to quickly generate accurate robot motion paths with the help of traditional path planning algorithms such as *PRM* and *RRT* methods. Moreover, the "width" and "height" of DT representation can be expanded by incorporating other attributes of the object, such as *mobility*, and

other scene information. However, several issues remain in this work: 1) the number of objects in scene is fixed, because the more target objects there are, the longer it takes for object detection and visual relationship recognition; 2) tasks and scenarios are kept relatively simple, with the number of atomic actions will not exceed 15 to prevent longer inference times; 3) there was a lack of subjective evaluation in robot control experiment.

It should be clarified that in the system proposed in our previous work [46], the task replanning (Section V) is preliminarily introduced. Compared to [46], this work has the following significant improvements. Firstly, a statistical analysis is performed on the task replanning dataset to verify the rationality of the labeled data. Secondly, labeled data of two new tasks are added to the original task replanning dataset to verify the generalization ability of the proposed model. Thirdly, a more in-depth analysis and discussion are conducted on the proposed task replanning model, and comparisons of different methods in task planning efficiency are added to verify the inference efficiency of the model. Lastly and most importantly, an RM-DT assisted human- robot control method is proposed and extensively validated through experiments, which horizontally expands the functionality of the system proposed in [46] and improves its adaptability to complex and uncertain manipulation tasks.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Grant No. 62088101), the Shanghai Rising-Star Program (23YF1449700), the National Natural Science Foundation of China (61825303, 52002286, 5197541, 620881015, U2013602), the Shanghai Municipal Science and Technology Major Project (2021SHZDZX0100) and the Fundamental Research Funds for the Central Universities (22120210547).

REFERENCES

[1] P. Balatti, D. Kanoulas, N. G. Tsagarakis, and A. Ajoudani, “Towards robot interaction autonomy: Explore, identify, and interact,” Proc. Int. Conf. Robot. Autom. (ICRA), Montreal, QC, Canada, May 2019, pp. 9523–9529.

[2] A learning from demonstration framework for adaptive task and motion planning in varying package-to-order scenarios

[3] Yue. Y, P. Zheng et al., "state-of-the-art survey on Augmented Reality-assisted Digital Twin for futuristic human-centric industry transformation, " Robotics and Computer-Integrated Manufacturing, 2023, 81: 102515.

[4] H. I Lin, et al., “Development of an intelligent transformer insertion system using a robot arm,” Robotics and Computer-Integrated Manufacturing, 2018, 51: 209–221.

[5] M. Prats, D. Pobil, A.P., et al., “Robot Physical Interaction through the combination of Vision, Tactile and Force Feedback,” Springer Tracts in Advanced Robotics, vol 84. Springer, Berlin, Heidelberg.

[6] M. Han et al., "Reconstructing Interactive 3D Scenes by Panoptic Mapping and CAD Model Alignments," 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 2021: 12199-12206.

[7] H. Du, X. Yu, L. Zheng, “Learning object relation graph and tentative policy for visual navigation,” 2019 European Conference on Computer Vision (ECCV), Glasgow, Springer, 2020: 19-34.

[8] Z. Jiao, Y. Niu and Z. Zhang, "Sequential Manipulation Planning on Scene Graph," 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 2022: 8203-8210.

[9] Z. Jin, A. Liu, W. Zhang and L. Yu et al," A Learning Based Hierarchical Control Framework for Human–Robot Collaboration," IEEE Transactions on Automation Science and Engineering, 2023, 20(1): 506-517.

[10] C. Li, P. Zheng, Shufei Li, " AR-assisted digital twin-enabled robot collaborative manufacturing system with human-in-the-loop, " Robotics and Computer-Integrated Manufacturing, 2022, 76: 102321.

[11] J. Lipton, A. J Fay and D. Rus, " Baxter's Homunculus: Virtual reality spaces for teleoperation in manufacturing, " IEEE Robotics and Automation Letters, 2017, 3(1): 179-186

[12] R. Moccia, C. Iacono, B. Siciliano and F. Ficuciello, "Vision-Based Dynamic Virtual Fixtures for Tools Collision Avoidance in Robotic Surgery," IEEE Robotics and Automation Letters, 2020, 5(2): 1650-1655.

[13] M. Chu et al., "Multisensory Fusion, Haptic, and Visual Feedback Teleoperation System Under IoT Framework," IEEE Internet of Things Journal, 2022, 9(20): 19717-19727.

[14] S.A Niederer et al, "Scaling Digital Twins from the Artisanal to the Industrial," Nature Computational Science, 2021, 1(5): 313-320.

[15] L. Zhang, L. Zhou, “Building a right digital twin with model engineering,” Journal of Manufacturing Systems, 2021, 59: 151-164.

[16] J. Liu, Z. Xu, H. Xiong, Q. Lin, W. Xu and Z. Zhou, "Digital Twin-driven Robotic Disassembly Sequence Dynamic Planning under Uncertain Missing Condition," IEEE Transactions on Industrial Informatics, 2023, 10.1109/TII.2023.3253187.
[17] D. Lee, SH L and et al., " Digital twin-driven deep reinforcement learning for adaptive task allocation in robotic construction, " Advanced Engineering Informatics, 2022, 53: 101710.
[18] WN Wang, W Ding and et al., "A digital twin for 3D path planning of large-span curved-arm gantry robot, " Robotics and Computer-Integrated Manufacturing, 2022, 76: 102330.
[19] X. Li, X. He B and et al., " Multisource model-driven digital twin system of robotic assembly, IEEE Systems Journal, 2021, 15(1): 114-123.
[20] X. Li, B. He, Z and et al., "Semantic-Enhanced Digital Twin System for Robot–Environment Interaction Monitoring," IEEE Transactions on Instrumentation and Measurement, 2021, 70: 7502113.
[21] Zhou Z, Yang X, Wang H, et al. Digital Twin with Integrated Robot-Human/Environment Interaction Dynamics for an Industrial Mobile Manipulator. 2022 International Conference on Robotics and Automation (ICRA). IEEE, 2022: 5041-5047.
[22] Mortlock T, Muthirayan D, Yu S Y, et al. Graph learning for cognitive digital twins in manufacturing systems. IEEE Transactions on Emerging Topics in Computing, 2021, 10(1): 34-45.
[23] Yao, B., Xu, W., Shen, T. et al. Digital twin-based multi-level task rescheduling for robotic assembly line. Sci Rep 13, 1769 (2023). https://doi.org/10.1038/s41598-023-28630-z
[24] Wang H, Li H, Wen X, et al. Unified modeling for digital twin of a knowledge-based system design[J]. Robotics and Computer-Integrated Manufacturing, 2021, 68: 102074.
[25] S. Cambon, R. Alami, and F. Gravot, “A hybrid approach to intricate motion, manipulation and task planning,” The International Journal of Robotics Research, 2009, 28 (1): 1-15.
[26] J. Sung, B. Selman, and A. Saxena, “Learning sequences of controllers for complex manipulation tasks,” in Proceedings of the International Conference on Machine Learning, 2013.
[27] M. Han, Z. Zhang, Z. Jiao et al, “Scene Reconstruction with Functional Objects for Robot Autonomy,” International Journal of Computer Vision, 2022, 130: 2940-2961.
[28] R. Martins, D. Bersan, M. Campos, et al, “Extending maps with semantic and contextual object information for robot navigation: a learning-based framework using visual and depth cues,” Journal of Intelligent and Robotic Systems, 2020, 99: 555-569.
[29] Deitke M, Han W and Herrasti A et al. RoboTHOR: An Open Simulation-to-Real Embodied AI Platform, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), USA, IEEE, 2020: 3161-3171.
[30] T. Chen, R. Chen, L. Nie, X. Luo, X. Liu and L. Lin, “Neural Task Planning With AND–OR Graph Representations,” IEEE Transactions on Multimedia, 2019, 21(4): 1022-1034.
[31] Z. Yu, J. Zhang, S. Mao et al, "RIRL: A Recurrent Imitation and Reinforcement Learning Method for Long-Horizon Robotic Tasks," 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 2022: 230-235.
[32] Eppe M, Nguyen P D H, Wermter S. From semantics to execution: Integrating action planning with reinforcement learning for robotic causal problem-solving[J]. Frontiers in Robotics and AI, 2019, 6: 123.
[33] Q. Wang, W. Jiao, P. Wang et al, “Digital twin for human-robot interactive welding and welder behavior analysis, “IEEE/CAA Journal of Automatica Sinica, 2020, 8(2): 1-10
[34] X. Wu, C. Yang, Y. Zhu et al, “An integrated vision-based system for efficient robot arm teleoperation, “Industrial Robot, 2020, 48(1):199-210.
[35] R. Li, H. Wang and Z. Liu, "Survey on Mapping Human Hand Motion to Robotic Hands for Teleoperation," IEEE Transactions on Circuits and Systems for Video Technology, 2022, 32(5): 2647-2665.
[36] L. Dominjon, A. Lécuyer, J. M. Burkhardt et al, " The "Bubble" Technique: Interacting with Large Virtual Environments Using Haptic Devices with Limited Workspace," First Joint Euro haptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. IEEE, 2005: 639–640.
[37] M. Kapteyn and J. Pretorius, " A Probabilistic Graphical Model Foundation for Enabling Predictive Digital Twins at Scale, " Nature Computational Science, 2021, 1(5): 337-347.
[38] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, “Digital twin in industry: State-of-the-art,” IEEE Transactions on Industrial Informatics, 2019, 15(4): 2405-2415.
[39] Yan Y, Tong L, Song K, et al, " SISG-Net: Simultaneous instance segmentation and grasp detection for robot grasp in clutter, " Advanced Engineering Informatics, 2023, 58: 102189.
[40] Pauwels P, de Koning R, Hendrikx B, et al, " Live semantic data from building digital twins for robot navigation: Overview of data transfer methods, " Advanced Engineering Informatics, 2023, 56: 101959.
[41] F. Tao, M. Zhang, Y. Liu, N.Y.C. Nee, “Digital twin driven prognostics and health management for complex equipment,” CIRP Annals, 2018, 67(1): 169–172.
[42] Y. Mo, S. Ma, H Gong, et al, “Terra: A smart and sensible digital twin framework for robust robot deployment in challenging environments,” IEEE Internet of Things Journal, 2021, 8(18): 14039-14050.

[43] A. Haddadi and K. Hashtrudi-Zaad, “Real-time identification of hunt–crossley dynamic models of contact environments,” IEEE Transaction on Robot, 2012, 28(3).
[44] P. W. Battaglia, J. B. Hamricka and Bapst V et al, “Relational inductive biases, deep learning, and graph networks, “in 2018 International Conference on Learning Representations (ICLR), Vancouver, 2018:1-8.
[45] B. Lacevic and P. Rocco, “Kinetostatic danger field-a novel safety assessment for human-robot interaction," in 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, IEEE, 2010: 2169-2174.
[46] X. Li, B. He, Z. Wang, Y. Zhou and G. Li, "Digital Twin-Driven Task Replanning Method for Robot-Environment Physical Interaction," 2022 12th International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), Baishan, China, 2022, pp. 407-412, doi: 10.1109/CYBER55403.2022.9907504.

APPENDIX

A. Application Method of Alita

Due to space limitations, some of the contents have not been detailed here, and the corresponding information of Alita can be obtained from the links in the Table VI.

TABLE VI Application Method of Alita

Item	Brief description			Reference link
RM-DT	Four-layer structure	Geometric model	Incremental ICP for surrounding reconstruction; URDF model for virtual robot reconstruction	Doi: 10.1109/JSYST.2019.2958874
		Physics model	Kelvin–Voigt model for contact dynamics modeling; modified mesh-based deformation method for object deformation modeling	Doi: 10.1109/JSYST.2019.2958874
		Relation model	Depth information aided visuospatial relationship detection framework (VRD-REI)	Doi: 10.1109/TIM.2021.3066542; https://github.com/forbiddenname/SEDTM-REI
		Semantic model	Semantic fusion and inference	Doi: 10.1109/TIM.2021.3066542
Task Re-Planning	Scene graph based Gns-lstm framework			Section V; https://github.com/forbiddenname/ DT-Alita
Dataset	Visuospatial dataset		5663 pairs of relationships annotations; 11 typical spatial relationships	Doi: 10.1109/TIM.2021.3066542; https://github.com/forbiddenname/SEDTM-RE
	Task planning dataset		4615 pairs of atomic actions annotations; 6 typical kitchen desktop tasks	Section V https://github.com/forbiddenname/ DT-Alita
Robot control method	Including position, joint, and command control			https://github.com/forbiddenname/ DT-Alita

B. From atomic action to robotic arm action

Fig. 15 URScript script representation of atomic actions.

In order to attach the actions to the motions, the atomic actions are represented by different URScript scripts (execution file of UR controller) according to its characteristics. When an atomic action is need to be executed, the corresponding script will be passed to the robot UR controller (using the official library *Python-urx*) from twin server, and then the robot executes the actions in the script, so as to control the robot through atomic action. The URScript script representations for different atomic actions are shown in Fig. 15. First, the wrist camera is used to locate the target object, and the path close to the target object is planned based on MoveJ to complete the *approach* operation; Then the *grab* point of the object is defined based on MoveL, and the opening and closing distance of the gripper are controlled to complete the *grab* operation. Similarly, *move together* and *place* operations can also be realized through these instructions; The *hold* operation can be expressed as keeping the grasping action until it changes; The operation of *pour into* and *stir* can be completed through joint angle 4 and joint angle 6. It should be noted that the above benefits from the Robotiq's wrist camera as it has the advantage of accurately identifying and tracking objects, and is compatible with UR robot.

C. Detailed structure of GNs-LSTM

Table VII and Fig. 16 represent the specific structure of the task replanning model, as well as the loss and accuracy curves during the training process, respectively. It can be seen that as the data iterates, the network gradually summarizes the characteristics of the planning model and after about 100 times, the network basically converges.

TABLE VII Detailed structure of GNs-LSTM

Step	Input	Operation	Details
(0)	Task Vector	Task features encode	(1) GRU(300, 64, batch_first=True, bidirectional=True), weights=((192, 300), (192, 64), (192,), (192,), (192, 300), (192, 64), (192,), (192,)); (2) Linear(1800, 640,), ReLU()
(1)	Scene graph X_E, X_S	Graph features encode1	(1) Linear(267, 512), ReLU(), Linear(512, 11); (2)(node_mlp1): Linear(139, 512), ReLU(), Linear(512, 139); (3) (node_mlp2): Linear(267, 512), ReLU(), Linear(512, 128)
(2)	(1)	Graph features encode1	(1) (edge_mlp1): Linear(267, 512), ReLU(), Linear(512, 11,); (2) (node_mlp1): Linear(139, 512), ReLU(), Linear(512, 139); (3) (node_mlp2): Linear(267, 512), ReLU(), Linear(512, 64)
(3)	(0), (2)	Sequence prediction	LSTM(22, 1280, batch_first=True), weights=((5120, 22), (5120, 1280), (5120,), (5120,))
(4)	(3)	Actions prediction	Linear(1280, 10), Softmax(), dropout=0.4
(5)	(3)	Objects prediction	Linear(1280, 11), Softmax(), dropout=0.4

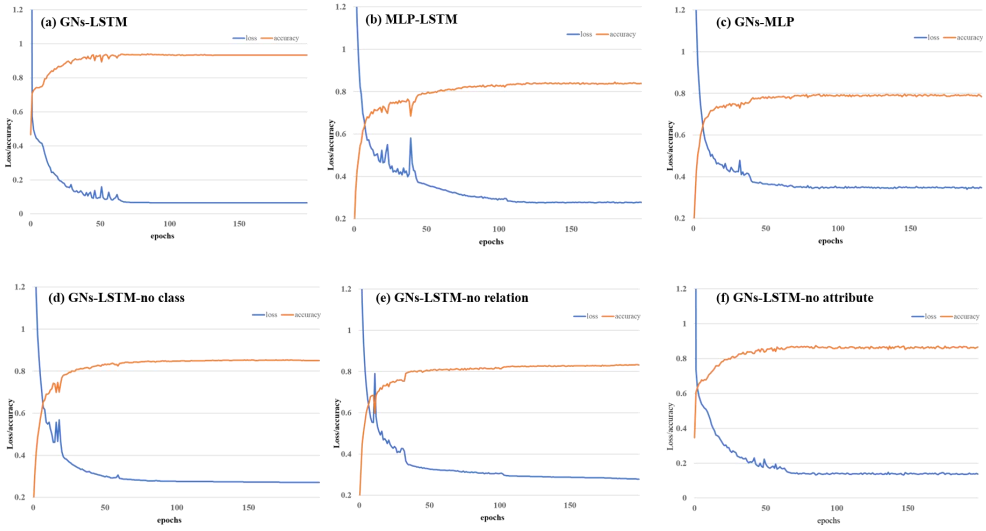

Fig. 16 Loss and accuracy curves during the training process.