Local Observation Based Reactive Temporal Logic Planning of Human-Robot Systems

Zhangli Zhou, Shaochen Wang, *Graduate Student Member, IEEE*, Ziyang Chen, Mingyu Cai, Hao Wang, Zhijun Li, *Fellow, IEEE*, and Zhen Kan, *Senior Member, IEEE*

***Abstract*— Human-robot collaboration plays an important role in intelligent manufacturing. However, the main challenge is how the robot can make online reactive changes to the plan based on the observed human behavior to ensure the completion of user-defined tasks. Such a challenge is further exacerbated if eye-in-hand manipulation is considered since the local field of the camera view cannot capture global observations. Different from existing planning approaches that separate the perception and planning modules, and make strong assumptions about perception abilities, we develop a framework of real-time local reactive planning that enables the robot to quickly adapt its actions if necessary through its limited perception of surroundings using an eye-in-hand camera. Specifically, we develop a locally observable transition system (LOTS) and interpretably express the task using linear temporal logic (LTL). To improve the grasping performance using local visual perception, we propose a high-resolution grasp network (HRG-Net) that achieves state-of-the-art results on multiple datasets (99.50% in Cornell and 97.50% in Jacquard and 96% in Graspnet-1Billion) for the task. A physical experiment using a 7DoF Franka Emika Panda robot demonstrates the effectiveness of the reactive planning framework.**

***Note to Practitioners*—Intelligent manufacturing often requires the human operator to work collaboratively with the robot in a shared workspace. Due to possible (assistive or non-assistive) interference of human operators, it is highly desired that the robot can perceive human behaviors and react properly to ensure task accomplishment. Hence, this work is particularly motivated to develop a reactive planning framework that relies on real-time local visual perception (i.e., eye-in-hand camera) to quickly react to its dynamic surroundings and replan its motion when necessary. In future work, rather than using the observed human behavior, we will investigate how to predict human intentions to further improve human-robot collaboration.**

***Index Terms*— Reactive planning, linear temporal logic, human-robot collaboration, intelligent manufacturing.**

I. Introduction

COLLABORATIVE robots engage increasingly in various applications ranging from intelligent manufacturing to more complex tasks, such as assistive robots in nursing homes or restaurants, flexible reconfigurable production in intelligent manufacturing, or even collaborative manipulation in outer space [1], [2], [3], [4], [5], [6]. In these applications, robots team up with human operators and work collaboratively in a shared workspace. To achieve intuitive and seamless human-robot collaboration, several challenges need to be tackled. Firstly, unlike conventional simple robotic manipulation tasks (e.g., point-to-point navigation), practical collaborative tasks can be complex and often involve a sequence of logically and temporally structured manipulations. For instance, mechanical parts have to be picked and installed sequentially by the robot with the assistance of human operators in the assembly line. Additionally, the challenge of complex tasks is further exacerbated when they are performed in a dynamic and contextually rich environment. Human behaviors, whether assistive or non-assistive, can change the surrounding environment. Unexpected environmental events or human-robot-environment interactions can also result in a dynamic environment. An example scenario is shown in Fig. 1. Hence, robots cannot always count on human teammates to stay on script. The robot must instead perform the task reactively, i.e., decide on the next action quickly as it observes the changes in the workspace. Appropriate reactive behaviors can complement and improve the performance of collaborative partners.

One common strategy to address the above challenges is reactive synthesis, which utilizes a replanning-and-execution method to adjust control strategies and adapt to changes in the environment. Previous studies [7], [8], [9], [10] have demonstrated that traditional reactive synthesis techniques rely on complete environmental observability to construct a transition system. Such a system models the interaction between a human and a robot as a two-player game and then identifies a winning strategy. However, despite its theoretical soundness, this approach is not practical in numerous intelligent manufacturing scenarios, where the dynamic environment often limits perception to a small amount of local information. For instance, as depicted in Fig. 1, even if a high-resolution

Manuscript received 30 November 2022; revised 17 April 2023 and 22 June 2023; accepted 6 August 2023. This article was recommended for publication by Associate Editor C. Lv and Editor M. Zhou upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant U2013601, Grant 62173314, Grant 62133013, and Grant U22A2060; in part by the National Key Research and Development Program of China under Grant 2022YFB4701400/4701403; in part by the Anhui Provincial Natural Science Foundation; and in part by the Anhui Energy-Internet Joint Program under Grant 2008085UD01. *(Corresponding author: Zhen Kan.)*

Zhangli Zhou, Shaochen Wang, Ziyang Chen, Hao Wang, and Zhen Kan are with the Department of Automation, University of Science and Technology of China, Hefei 230026, China (e-mail: zkan@ustc.edu.cn).

Mingyu Cai is with the Department of Mechanical Engineering, University of California at Riverside, Riverside, CA 92521 USA (e-mail: mingyu.cai@ucr.edu).

Zhijun Li is with the Department of Automation, University of Science and Technology of China, Hefei 230026, China, and also with the School of Mechanical Engineering, Tongji University, Shanghai 201804, China.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASE.2023.3304842.

Digital Object Identifier 10.1109/TASE.2023.3304842

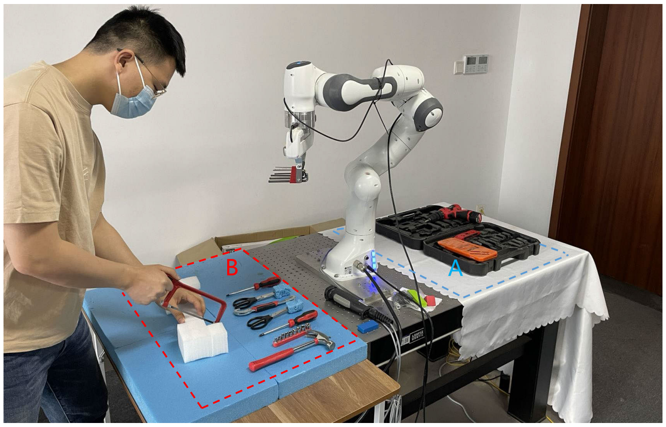


Fig. 1. The robot works collaboratively with a human partner in a shared workspace. The robot is desired to select an appropriate tool from the source workspace A, hand it to the human, and assist the human in the goal workspace B. Throughout this collaborative task, humans may occasionally step in to assist the robot, such as helping them with part of their work or providing feedback. Unfortunately, human errors can also occur, such as misplacing tools. Due to these possible interactions with humans, the robot needs to perceive its surroundings and quickly react and replans its actions whenever something does go off-script.

camera is strategically placed using the eye-to-hand technique to achieve global workspace observability, it is still likely to be obstructed by the robotic manipulator during the manipulation process, thereby making global observability unattainable.

To deal with the aforementioned issues, in this framework, the camera is installed at the end-effector of the manipulator, and the eye-in-hand approach is utilized to enable more precise predictions of the grasp position and orientation as the arm approaches the target object. To address the issue of local observability during motion planning, we developed a high-resolution grasping network (HRG-Net) that integrates multi-modal inputs and different scales of feature map information. This enables the use of the original-sized feature map to achieve precise high-resolution representation, while simultaneously downsampling to obtain a richer semantic representation. Our experiments on the Cornell and Jacquard datasets demonstrate that the HRG-Net outperforms existing methods. Significant improvements have been made over our previous work [11]. Firstly, we conducted tests on the GraspNet-1Billion [12] dataset and achieved state-of-the-art results. Secondly, we tested the algorithm on real physical scenarios involving dynamic objects and object generalization with random initial positions. Our results show that the HRG-Net is superior to other methods, achieving time savings of approximately $9\% - 21\%$ when performing the same task. This is primarily due to the HRG-Net's ability to generate more accurate predictions of the grasping position and orientation while avoiding time-consuming exploration during motion planning.

To enable reactive synthesis based on local vision-based observation of the environment, we propose a novel locally observable transition system (LOTS), which associates the robot's perceived environment states with the task progress. Building this association requires a structured representation of a task so that the task progress (i.e., the completion of subtasks) can be verified online according to the perceived environment states. We apply linear temporal logic (LTL) [13] to express high-level tasks, which can model a rich class of human-interpretable tasks for robots. Its crucial benefit is to employ an automaton to track the process of satisfaction. This allows us to utilize the environment states and the robot states to conveniently check the automaton transitions through the perceived information, based on which the robot can replan correspondingly toward the completion of the task. In this framework, our approach can utilize reactive synthesis methods in a dynamic environment, which removes the requirement of global observability in previous reactive synthesis methods [7], [8], [9]. Furthermore, since the size of the product automaton can grow exponentially large as the environment size and the complexity of the LTL task increases, searching all feasible paths in the product automaton is computationally expensive. Instead, we establish a mapping from automata to LOTS through the visual perception that can conduct fast reactive planning by searching only in the automaton. It is shown that our approach has a lower space complexity ranging from $O((n+1)^2)$ to $O(n^4)$ in the worst case, compared to the traditional reactive synthesis with the complex range of $O((n+1)^2 m^2)$ to $O(n^4 m^2)$.

The main contributions are summarized as follows:

- We propose a paradigm for human-robot interactions to accomplish high-level complex tasks relying on local perception in dynamic unstructured environments.
- We relax the restriction of requiring a fully observable environment in [7], [8], and [9] and achieve fast reactive planning with reduced algorithm complexity.
- In the perception module, we extend the preliminary work [11] and develop a high-resolution grasping network (HRG-Net) that fuses multi-modal inputs to achieve safe and reliable closed-loop control in dynamic environments, achieving seamless human-robot collaboration. Physical demonstration using a 7 DoF Franka Emika Panda robot shows the effectiveness of the developed reactive task and motion planning framework.

II. Related Works

A. Reactive Planning

Traditional motion planning refers to the control strategy that a robot maps directly from its state to its available actions. However, this strategy is likely to fail when the environment changes. To enable reactive behaviors, joint perception, and planning have been widely investigated for robotic systems. For instance, Wang et al. [14] introduce a framework that leverages predictive modeling of human behavior to facilitate effective collaboration between humans and machines in complex tasks. Raessa et al. [15] presented a constraint-based incremental manipulation planning method that generates robot motions to facilitate efficient and comfortable human-robot collaboration in assembly tasks. The previous work [14], [15] discussed the attainment of efficient collaboration through reactive planning in structured environments. In contrast, this paper addresses the issue of reactive planning when relying solely on local perception in unstructured and dynamic environments. Bai et al. [16] proposed the partially observable Markov decision process to integrate perception and planning in the continuous space. Ghosh et al. [17] presented an efficient obstacle avoidance approach based on joint perception and planning using stereo vision. Seraj et al. [18] developed, a collaborative planning and control algorithm to enhance the cooperative behaviors of a multi-robot system for joint perception-action tasks in dynamic environments. Despite the progress, the works of [16], [17], and [18] mainly focus on simple tasks, such as point-to-point navigation or obstacle avoidance, and the developed joint perception and planning approaches cannot be immediately extended to handle complex tasks that involve a series of logically organized actions.

Due to the rich expressivity of temporal logic in specifying logically organized actions, reactive synthesis with temporal logic specifications has attracted growing research attention. He et al. [8], [9] considered a reactive strategy synthesis with resource-constrained finite tasks and developed a compositional approach, which achieves orders-of-magnitude speed-up over existing explicit approaches for pick-and-place tasks. Ghasemi et al. [19] developed an active perception and planning algorithm for the agent to realize the task with high probability in an environment with partially known semantics. Li et al. [7] developed a dynamically reconfigurable planning methodology using behavior tree-based control strategies, which shows efficient recovery functionalities with a minimal number of replanning steps. Well et al. [10] considered probabilistic reactive synthesis for collaborative human-robotic manipulation tasks. Since pre-specified tasks can become infeasible in a dynamic environment, tasks with temporal logic specifications are often relaxed to enable reactive motion replanning. For instance, least-violating control strategies were developed in the works [20], [21], [22], [23]. In the work [24], [25], [26], hard and soft constraints were considered where the soft constraints can be violated to allow motion replanning. Other representative results include receding horizon control based reactive planning [27], [28], learning-based approaches [29], [30], and sampling-based reactive methods [31], [32], [33]. Despite the progress, the aforementioned works either rely on a product graph for motion planning, which can become intractable if the task and environment complexity is high, or assume global observability, i.e., global awareness of the environment and robot states. Limiting to using a "third-person" camera with global observability, these approaches do not readily apply to the eye-in-hand manipulation considered in this work.

B. Parallel Gripper Robot Grasping

Parallel jaw robotic grasping detects oriented rectangles in the image, which represent promising grasp candidates for parallel jaw grippers. The ability to locate the object's position and determine the appropriate grasping pose is crucial to stable and robust robotic grasping. Earlier works [34], [35] were mainly geometry-driven, which requires the knowledge of the geometric model of the grasping object, limiting its generalization capability to new objects. The data-driven approaches [36], [37], [38], [39] are well positioned to learn the grasp paradigm implicitly embedded within the data. Prior works were to extract a series of bounding box proposals and select the best one as output through a two-stage network. However, such an approach requires a large number of candidates, which is time-consuming. Later many works [38], [40], [41], [42] utilize convolutional neural networks to generate bounding box proposals to estimate the grasp pose of objects. However, these methods all rely on an encoder-decoder structure, which indicates that spatial information is compromised. Although spatial information can be recovered by upsampling, the predicted results may fail in practice, especially when there are multiple objects sticking together or the camera is far away from the target which yields compound pixel errors. Previous methods overemphasize the high-level semantic information and ignore the low-level spatial information. Therefore, the high-resolution feature maps are retained in our HRG-Net to mitigate the loss of spatial information.

III. Problem Formulation

A. Preliminary Background

Definition 1: **Linear temporal logic (LTL)** is a formal logic whose basic component is a set of atomic propositions Π. The standard Boolean operators such as $\wedge$ (conjunction), $\vee$ (disjunction), $\neg$ (negation), and temporal operators such as $\Diamond$ (eventually), $\bigcirc$ (next), $\square$ (always), and $\mathcal{U}$ (until). The syntax of an LTL formula ϕ is defined as:

$$\phi ::= true \mid p \mid \phi_1 \wedge \phi_2 \mid \neg\phi_1 \mid \bigcirc\phi \mid \phi_1 \mathcal{U} \phi_2 \ ,$$

where $p \in \Pi$ is an atomic proposition, *true*, *negation* $\neg$, and *conjunction* $\wedge$ are propositional logic operators, and *next* $\bigcirc$ and *until* $\mathcal{U}$ are temporal operators. $\square\phi$ means ϕ is true for all future moments; $\Diamond\phi$ means ϕ is true at some future moments; $\bigcirc\phi$ means ϕ is true at the next moment; and $\phi_1\mathcal{U}\phi_2$ means ϕ_1 is true until ϕ_2 becomes true.

The semantics of an LTL formula are defined over an infinite sequence $\sigma = \sigma_0\sigma_1 \ldots$ with $\sigma_i \in 2^\Pi$ for all $i \geq 0$, where 2^Π represents the power set of Π. In summary, the semantics of LTL are defined as:

$$\sigma \models true$$
$$\sigma \models p \Leftrightarrow p \in \sigma(0)$$
$$\sigma \models \phi_1 \wedge \phi_2 \Leftrightarrow \sigma \models \phi_1 \text{ and } \sigma \models \phi_2$$
$$\sigma \models \neg\phi \Leftrightarrow o \not\models \phi$$
$$\sigma \models \bigcirc\phi \Leftrightarrow o[1:] \models \phi$$
$$\sigma \models \phi_1\mathcal{U}\phi_2 \Leftrightarrow \exists t \geq 0 \text{ s.t. } \sigma[t:] \models \phi_2, \forall t' \in [0, t), \sigma[t':] \models \phi_1$$

More detailed descriptions of the syntax and semantics of LTL can be found in [43]. An LTL formula can be translated to a nondeterministic Büchi automaton (NBA).

Definition 2: **Nondeterministic Büchi Automaton (NBA)** $\mathcal{B}$ is a tuple $\mathcal{B} = (Q, Q_0, \Sigma, \delta, F)$, where Q is a finite set of states, $Q_0 \subseteq Q$ is a set of initial states, $\Sigma = 2^\Pi$ is the alphabet from LTL formula, $\delta : Q \times \Sigma \to 2^Q$ is a non-deterministic transition function, and $F \subseteq Q$ is a set of accepting states called the acceptance set.

Let $q \overset{\sigma}{\to} q'$ denote the transition from $q \in Q$ to $q' \in Q$ under the input $\sigma \in \Sigma^w$ iff $q' \in \delta(q, \sigma)$. Given a sequence of input $\boldsymbol{\sigma} = \sigma_0\sigma_1\sigma_2\ldots$ over Σ^w, the generated infinite sequence $\boldsymbol{q} = q_0q_1q_2\cdots$ is a run of $\mathcal{B}$, where $q_i \xrightarrow{\sigma_i} q_{i+1}$ for $i \geq 0$. Run $q_0q_1q_2\ldots$ is accepting if $q_i \in F$ for infinitely many indices $i \in \mathbb{N}$.

Due to the consideration of eye-in-hand manipulator, the field-of-view (FOV) of the camera is time-varying as the end-effector moves, resulting in limited observability of environment states. Specifically, in the model building example, the robot can only observe at most half of the workspace (e.g., either the A or B in Fig. 1). Hence, we consider only task-related objects in the workspace and introduce a local observable transition system (LOTS) to record the task progress.

Definition 3: **Locally Observable Transition System (LOTS)** $\mathcal{T} = (S, s_0, A_t, \delta_t, \mathcal{O}, \Pi, \mathcal{L})$ where $S = S^A \cup S^B$ is the finite set of states consisting of S^A and S^B that represent all possible states of W^A and W^B respectively (We divide the whole workspace W into two rectangular areas W^A and W^B as shown in Fig. 1, where W^A denotes source workspace, and W^B denotes goal workspace), s_0 is an initial state, A_t is a set of actions, $\delta_t : S \times A_t \to S$ is a deterministic transition, Π is a set of atomic propositions, $\mathcal{O}$ is the state of the part of the workspace that the robot can observe and $\mathcal{L} : S \to 2^{\Pi}$ is a labeling function that maps the state to a subset of atomic propositions that hold true.

Suppose there are M classes of objects $R = (o_1, o_2, \ldots, o_M)$ (e.g., hammer, knife, etc.) in workspace W and let $s = o_i^n$, $i \in \{1, \ldots, M\}$ denotes the number of o_i as n. The class and number of task-related objects in W^A and W^B are denoted by s^A and s^B, respectively, which gives rise to the state s in LOTS. For instance, if there are multiple classes of objects in the local FOV of the camera, the state in $\mathcal{T}$ is denoted as $s = s^A\_s^B$ with $s^A = o_1^1o_2^3o_3^2$ and $s^B = o_1^0o_2^0o_3^0$, which indicates that there are 1, 3, and 2 of o_1, o_2, o_3 in W^A and none of o_1, o_2, o_3 in W^B, respectively. The action set A_t consists of a series of task-related actions. For instance, the action $a_B^A(o_i) \in A_t$ is to pick the object o_i (e.g., a hammer) from W^A and then places it in W^B. An atomic proposition $\pi \in \Pi$ corresponding to a state $s \in S$ can be either true or false. For example, $\pi = o_i^1\_o_j^2$ is true if and only if there are one o_i in W^A and two $o_j^{\prime}s$ in W^B. $\mathcal{L}$ maps a state $s \in S$ to a set of true atomic propositions $\mathcal{L}(s) \subseteq \Pi$.

B. Problem Formulation

Consider a human and a robot collaborating to complete an intelligent manufacturing-related temporal logic task ϕ in a shared workspace W. To elaborate on the human-robot collaboration task, the model-building example introduced in Fig. 1 will be used as a running example throughout the work.[1] Specifically, suppose the workspace W consists of a source workspace W^A and a goal workspace W^B, where $W = W^A \cup W^B$ and $W^A \cap W^B = \emptyset$. The task of building a model is specified by the LTL formula ϕ. There is no direct communication between the robot and the human throughout the process. The robot's awareness of the workspace W relies only on the perception information of the RGB-D camera mounted at its end-effector. During model building, the human can interact with the robot in either an assistive (e.g., picking up a tool) or a non-assistive manner (e.g., misplacing tools or taking away tools). Specifically, we consider two classes of human interactions:

- dis_1: Human only moves objects between W^A and W^B, and thus the type and number of objects in $W = W^A \cup W^B$ will not change.
- dis_2: The type and number of objects in W change due to human behaviors (e.g., taking away objects from W or adding new objects to W).

Assumption 1: It's assumed that the given LTL specification can always be satisfied during the process of human-robot interactions.

Problem 1: Given an LTL task ϕ, the objective is to develop reactive motion planning and accomplish it using real-time local visual perception (i.e., eye-in-hand camera) under human-robot interactions i.e., dis_1 and dis_2.

Due to the consideration of interactions with the human operator, it is possible that the human actions can directly lead to task failure (e.g., the human operator discards all items in the workspace. As a result, the task may never be completed by the robot), or repeatedly place the items picked by the robot back to the workspace. Hence, Assumption 1 is made to ensure that the required task is theoretically possible to be completed.

Motivation: The attainment of effective human-robot collaboration in intelligent manufacturing is a critical concern. The current approaches often have limitations, such as necessitating robots to possess comprehensive knowledge of the entire process or mandating a fixed task assignment between humans and robots. While the former is difficult to achieve in practical environments, the latter can only offer generic collaboration in assembly lines. Furthermore, as tasks become more intricate, human errors may result in the failure of collaborative tasks. To address these limitations, we present a framework that enables robots to perform collaborative tasks with humans, leveraging local perception and no fixed task assignments. To reflect the robot's reactive planning capabilities, we enable complex temporal logic constraints between various subtasks. Our framework aims to infer the state of the entire workspace by monitoring the state of the local workspace. By comprehending the human's impact on the collaborative task through the workspace state, the robot can select suitable actions to collaborate with the human and successfully complete the task. This approach empowers humans to utilize their creativity during collaborative tasks without undue constraints.

IV. Solution

This section presents the reactive planning framework, which consists of a perceptual and reactive task planning module and a motion planning module, as shown in Fig. 2. In addition, our framework involves human-robot collaboration, and we employ several mechanisms to ensure the security of this collaboration.

[1] The developed visual perception and reactive planning framework is not limited to collaborative manipulation tasks and can be extended to a variety task (e.g., persistent navigation) that involves visual feedback and a series of logically organized actions.

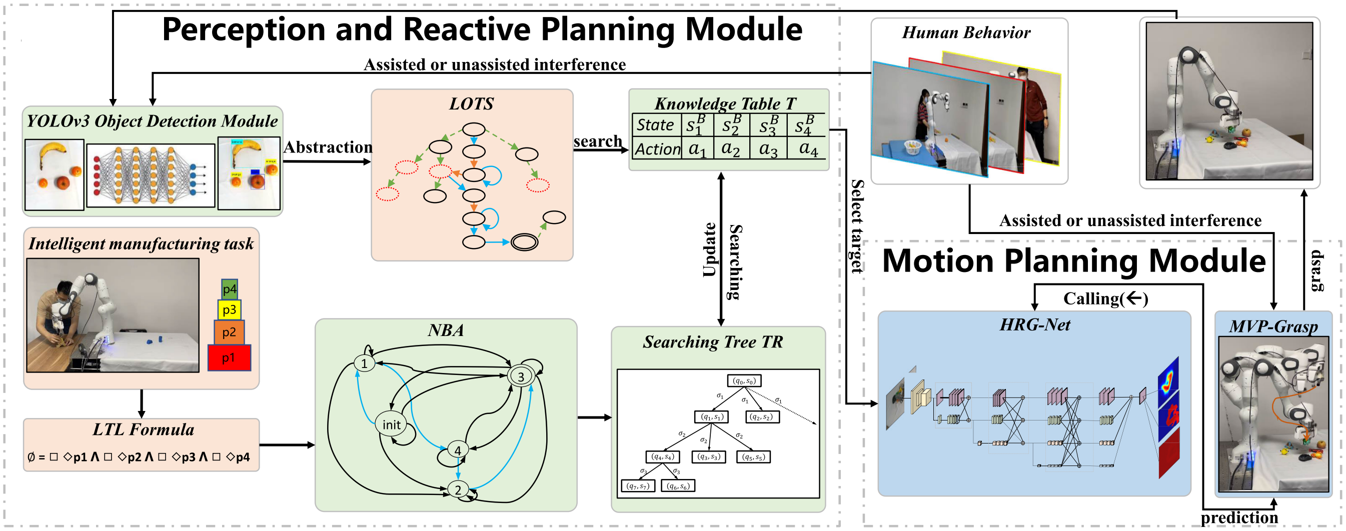

Perception and Reactive Planning Module
Assisted or unassisted interference
YOLOv3 Object Detection Module
Abstraction
LOTS
search
Knowledge Table T
State s_1^B s_2^B s_3^B s_4^B
Action a_1 a_2 a_3 a_4
Human Behavior
Select target
Update
Searching
Intelligent manufacturing task
p4
p3
p2
p1
Assisted or unassisted interference
Motion Planning Module
grasp
Calling(←)
HRG-Net
MVP-Grasp
LTL Formula
∅ = □ ◇p1 ∧ □ ◇p2 ∧ □ ◇p3 ∧ □ ◇p4
NBA
init
1
2
3
4
Searching Tree TR
prediction

Fig. 2. The overview of the real-time visual perception and reactive planning framework. The left describes the reactive planning of the task, which incorporates visual detection methods and automata theory to quickly determine the progress of the task using only local visual feedback and correctly select the next action in the presence of human interactions. On the right is a module on motion planning. A high-resolution grasping network is developed for robust and stable grasping in a dynamic and cluttered environment.

A. Perception Based Reactive Task Planning Module

The developed perceptual and reactive task planning module in this work contains two main components as shown in the left of Fig. 2. One is an LTL task ϕ that incorporates the intelligent manufacturing task, and the other is an online reactive task planning algorithm that combines perception and decision-making.

The visual perception and planning algorithm is outlined in Alg. 1. Given the LTL formula ϕ and the LOTS $\mathcal{T} = (S, s_0, A_t, \sigma_t, \mathcal{O}, \Pi, \mathcal{L})$, we first construct the corresponding NBA $\mathcal{B} = (Q, Q_0, \Sigma, \delta, F)$. The initial states $q_0 \in \mathcal{Q}_t$ and $s_0 \in \mathcal{S}$ are set as the root node of the search tree TR. (lines 1 - 2 in Alg. 1). The Dijkstra search algorithm is used to search for the shortest path from $q_0 \in Q_0$ to $q_F \in F$ over $\mathcal{B}$, which yields an accepting run $\boldsymbol{q} = q_0 q_1 q_2 \ldots, q_i \in Q$ of $\mathcal{B}$ and the associated state trajectory $\boldsymbol{s} = s_0 s_1 s_2 \ldots, s_i \in S$, since the action set A_t in $\mathcal{T}$ is defined over the alphabet Σ and the actions $a_i = a(\sigma_i) \in A_t$ are performed when $q_i \xrightarrow{\sigma_i} q_{i+1}$. The state-action pairs $(s_i^B, a_i), s_i^B \in S^B, a_i \in A_t$ and the state pairs $(q_i, s_i^B), q_i \in Q, s_i^B \in S$ are then recorded to build the initial table T and the search tree TR, respectively (Alg. 2). In this way, the current state $s_{current}^B \in S^B$ can be inferred from the observations of the camera, and the next action $a_{i+1} \in A_t$ to be taken can be determined by the knowledge base table T (lines 5 - 14 in Alg. 1). However, this offline knowledge database T can only quickly react to known states (i.e., encountered situations). If a new situation is encountered, Alg. 3 is invoked to update the search tree TR and thus update the knowledge database table T to account for the new situations. We will show later in Theorem 1 that there always exists at least one path in LOTS that satisfies the LTL formula ϕ so that we can establish a mapping from $\mathcal{B}$ to $\mathcal{T}$.

Due to possible interactions with human operators, the state may not transit as desired. For instance, a tool supposed to be picked up by the manipulator may be misplaced by the human operator, or a new object is added by the human operator to the workspace. The above two kinds of interference correspond to dis_1 and dis_2 in the III-B, both of them may lead to a sudden change in the state of $s \in S$. If Assumption 1 holds, the maximum possible state of S^B that depends on ϕ is a invariant with respect to human interactions, while maximum possible states of S^A may increase. We verify it as follows:

Lemma 1: Given the LTL task ϕ and the initial state of W is $s_0 = o_1^{n_1} o_2^{n_2} o_3^{n_3} \ldots o_x^{n_x} \_ o_1^0 o_2^0 o_3^0 \ldots o_x^0$, let $|S^B|$ denote the cardinality of the state set S^B. If Assumption 1 holds, $|S^B|$ is always equal to a constant C depending on ϕ under any disturbances dis_1 and dis_2.

Proof: Suppose that, under the LTL formula ϕ, the initial state of W is $s_0 = o_1^{n_1} o_2^{n_2} o_3^{n_3} \ldots o_x^{n_x} \_ o_1^0 o_2^0 o_3^0 \ldots o_x^0$ and the final state of W^B is $s_{final}^B = o_1^{n_1'} o_2^{n_2'} o_3^{n_3'} \ldots o_x^{n_x'}$, where $n_i' \geq 0, \forall i \geq 0$. Let $s_t^B = o_1^{t_1} o_2^{t_2} o_3^{t_3} \ldots o_x^{t_x}$ denote the state of S^B at the moment t that the task has not completed yet. According to Assumption 1, we have $t_i \leq n_i', \forall i > 0$. If $t_i > n_i'$, then there exists a state where the object o_i needs to be moved from W^B to W^A, which violates Assumption 1. Hence, there are at most n_i' objects o_i on W^B, and the cardinality of states S^B is $|S^B| = \prod_{i=1}^{x}(n_i'+1)$, i.e., $C = \prod_{i=1}^{x}(n_i'+1)$. ■

Suppose the current state $s_{current}^B \in S^B$ of W^B as inferred from local observations, the robot can then determine exactly the next action to be selected. When $s_{current}^B \in T$, we can select the appropriate action $a_i \in A_t$ according to the hash table T. If $s_{current}^B$ does not exist in T (e.g., the task does not proceed as desired due to human interactions or the environment changes), the tree TR is updated by Alg. 3. The general idea in Alg. 3 is to randomly select a node (q, s^B) in TR as the starting point and grow the tree by generating new pairs (q_{next}, s_{next}^B) until $s_{current}^B$ is reached. Once $s_{current}^B$ is reached, T is updated according to it's corresponding q, and continue to complete the task ϕ using T (lines 16 - 20 in Alg. 1). If the total number of edges in TR exceeds that of W^B, it means that the task ϕ cannot be completed from the current $s_{current}^B$ (lines 22 in Alg. 1).

Algorithm 1 Visual Perception and Planning

```
Input: The task LTL formula $\phi$, the LOTS
       $\mathcal{T} = (S, s_0, A_t, \sigma_t, \mathcal{O}, \Pi, \mathcal{L})$
Generate the corresponding $\mathcal{B} = (Q, Q_0, \Sigma, \delta, F)$;
Create a tree $TR$ with the root node $(q_0, s_0^B)$;
Construct a table $T$ = get_table($q_0, s_0^B, \mathcal{B}$) by Alg. 2;
while $True$ do
    Get $s_{lo}$ by local observation;
    if $s_{lo} \in S^A$ then
        $s^B_{current} = s \setminus s_{lo}, s^B = s^B_{current}$
    else
        $s^B_{current} = s_{lo}, s^A = s \setminus s_{lo}$
    end
    $s = s^A\_s^B$;
    if $s^B_{current}$ exists in $T$ then
        Determine $a_i$ such that $\left(s^B_{current}, a_i\right) \in T$;
        Perform $action = a_i$;
    else
        $TR$ = tree_update($\mathcal{B}, \mathcal{T}, TR, s^B_{current}$) by Alg.
         3;
        if $s^B_{current}$ exists in $TR$ then
            Find $q \in Q$ satisfying $(q, s^B_{current}) \in TR$;
            $T$ = get_table$\big(q, s^B_{current}, \mathcal{B}\big)$;
            Determine $a_i$ such that $\left(s^B_{current}, a_i\right) \in T$;
            Perform $action = a_i$;
        else
            Perform $action = wait$;
        end
    end
end
```

Algorithm 2 Function get_table()

```
Input: Automaton state $q_s$, LOTS state $s_s^B$, $\mathcal{B}$
Output: $T$
$\left[\boldsymbol{q}, \boldsymbol{\sigma}\right] = Dijkstra\left(q_s, s_s^B, \mathcal{B}\right)$ ;
for $i$=0:length($\boldsymbol{\sigma}$)-1 do
    $a_i = a(\boldsymbol{\sigma}[i])$;
    $s^B_{i+1} = \delta_t(s^B_i, a_i)$;
    Add $(s^B_i, a_i)$ to $T$;
    $q_i = \boldsymbol{q}[i]$;
    Add $(q_i, s^B_i)$ to $TR$;
end
return $T$;
```

Algorithm 3 Function tree_update()

```
Input: Automaton $\mathcal{B}$, $\mathcal{T}$, $TR$, $s^B_{current}$
Output: $TR$
while $s^B_{current} \notin TR$ do
    Randomly select a node $(q', (s^B)')$ from $TR$;
    for $\sigma_i \in \Sigma$ do
        $q_{next} = \delta(q', \sigma_i) \in Q$;
        $s^B_{next} = \delta_t((s^B)', a(\sigma_i))$;
        Add $(q_{next}, s^B_{next})$ to $TR$;
        if $|TR| \geq |W^B_{end}|$ then
            break;// $|TR|$ represents the
                number of elements in the
                $TR$, and $|W^B_{end}|$ represents
                the number of possible
                states of all LOTS in $W^B$
                before the task is
                completed.
        end
    end
end
return $TR$;
```

To elaborate the perceptual and reactive task planning framework, the following example is provided.

Example 1: Consider a manufacturing scenario that requires a set of tools, such as a hammer, screw and screwdriver. The fabrication process consists of a series of steps. For example a specific task can be expressed in the LTL formula as $\phi_{model} = \Box\Diamond p1 \wedge \Box\Diamond p2 \wedge \Box\Diamond p3 \wedge \Box\Diamond p4$, where $p1$ indicates the action of taking the hammer from W^A to W^B, $p2$ and $p4$ indicate taking the screws from W^A to W^B the first and the second time respectively, and $p3$ indicates taking the screwdriver from W^A to W^B. Initially, there are 1 hammer, 1 screwdriver, and 2 screws in the W^A, and one basket in W^B. Consequently, the initial state of LOTS is $S_0 = s_0^A\_s_0^B$, where $s_0^A = h^1s^2d^2$ and $s_0^B = h^0s^0d^0$ (where h represent hammer, s correspond screw and d present screwdriver). The goal is to put the items in the basket in the desired order according to ϕ_{model}. Given ϕ_{model}, the generated automaton and LOTS corresponding to the task ϕ_{model} are shown in Fig. 3 and Fig. 4. The video corresponding to the example is shown in the link https://youtu.be/O94KcVqwccA.

Under Assumption 1, Lemma 1 indicates that the robotic manipulation only needs to decide the progress of the task based on the current state of W^B, since dis_1 and dis_2 cannot trigger a new state for S^B. Based on Lemma 1, the following theorem shows that there exist at least one trajectory in $\mathcal{T}$ that satisfies ϕ.

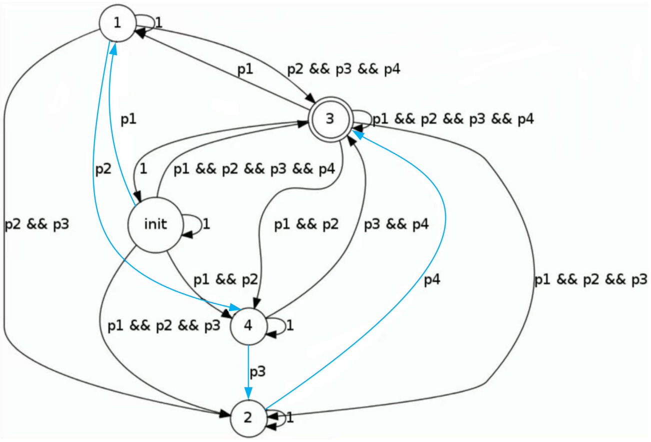


Fig. 3. The Büchi automaton $\mathcal{B}$ corresponding to the LTL task $\phi_{model} = \Box\Diamond p1 \wedge \Box\Diamond p2 \wedge \Box\Diamond p3 \wedge \Box\Diamond p4$. The blue arrow lines indicate a path that satisfies ϕ_{model}, which is mapped to a feasible path in LOTS of Fig. 4.

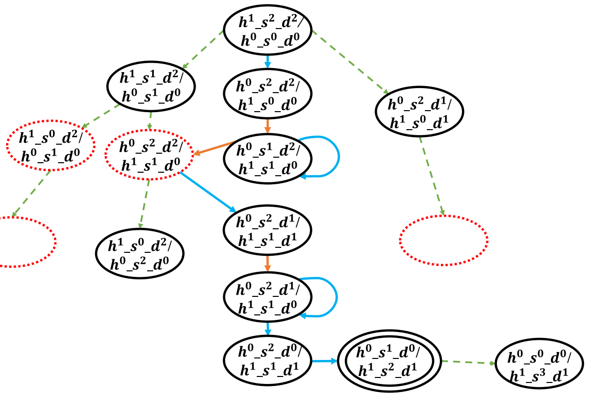


Fig. 4. The constructed LOTS, where the node represents its states and the edges represent potential transitions. The green dotted lines that connect the potentially reachable states and the red dashed boxes indicate the states that can be reached due to dis_2. The empty dashed ellipses indicate possible LOTS states that are not listed.

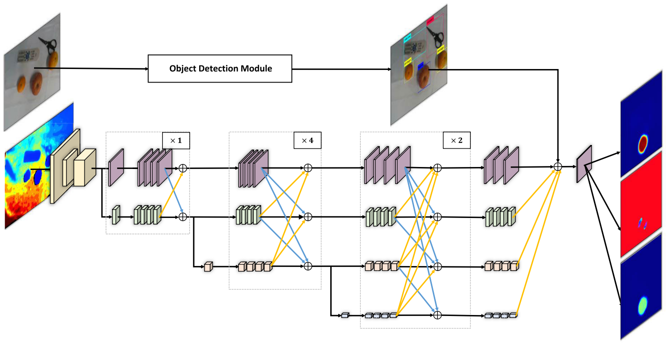


Fig. 5. The architecture of the HRG-Net. The high-resolution feature maps are maintained in the whole network while low-resolution features are gradually introduced and convolutions of different resolutions are connected in parallel. This enables information interchange between representations of different resolutions, which improves the spatial sensitivity while ensuring strong semantic expressiveness.

Theorem 1: Given a manipulation task specified by the LTL formula ϕ, if Assumption 1 holds, Alg. 1 can find at least one trajectory in $\mathcal{T}$ that satisfies ϕ.

Proof: Consider an LTL formula ϕ and its corresponding $\mathcal{B} = (Q, Q_0, \Sigma, \delta, F)$. Without loss of generality, suppose that initially there are x classes of task-related objects $o_1, o_2, o_3, \ldots, o_x$ with numbers $n_1, n_2, n_3, \ldots, n_x$, respectively. The initial state of the workspace W is $s_0 = o_1^{n_1} o_2^{n_2} o_3^{n_3} \ldots o_x^{n_x}\_o_1^0 o_2^0 o_3^0 \ldots o_x^0$. It is assumed that, under the LTL formula ϕ, the final state of the W^B is $s_{final}^B = o_1^{n_1'} o_2^{n_2'} o_3^{n_3'} \ldots o_x^{n_x'}$ $(n_i' \geq 0, \forall i \geq 0)$. Under Assumption 1, we can learn from Lemma 1, the number of all possible task-related states in W^B is equal to $\prod_{i=1}^{x}(n_i'+1)$. Let $q_0 \in Q_0$ and $s_0^B = o_1^0 o_2^0 o_3^0 \ldots o_x^0$ are the initial state of $\mathcal{B}$ and W^B, respectively. By the construction and update of TR and the probabilistic completeness of the sampling-based method [44], all possible input sequence $\boldsymbol{\sigma}$ and the corresponding trajectory $\boldsymbol{q}$ that satisfies ϕ can be generated in TR. In other words, for $\forall q \in Q$, there always exists an input sequence $\boldsymbol{\sigma} = \sigma_0 \sigma_1 \ldots \sigma_i$ that yields a trajectory from q to $q_F \in F$. Since the action set A_t in $\mathcal{T}$ is defined over Σ, when $a_i = a(\sigma_i)$ is executed, the state transition in $\mathcal{T}$ occurs. Note that, regardless of the interactions, the state $s_t^B = o_1^{n_{t_1}} o_2^{n_{t_2}} o_3^{n_{t_3}} \ldots o_x^{n_{t_x}}$ always satisfies $n_{t_i} \leq n_i', \forall i \geq 0$. Since S^B contains all possible states in W^B, we can definitely find a $s^B \in S^B$ corresponding to $q \in Q$. Considering that S^A is changing, there may exists $s_1, s_2 \in S, st.s_1^B = s_2^B$ but $s_1 \neq s_2$. Therefore, there always exists at least a path in $\mathcal{T}$ corresponding to $\boldsymbol{q}$ that satisfies ϕ. ■

Theorem 1 indicates the desired task ϕ can be guaranteed to be completed by identifying and following a corresponding path in LOTS. Compared with traditional approaches [7], [8], [9], [10], we have the following benefits. First, our approach relies only on local perception and does not require global observability. Second, our method searches directly in the automaton, avoiding the need of building a product automaton and searching for paths in the product automaton. The following lemma shows that the complexity of our approach is significantly reduced.

Lemma 2: The complexity of LOTS-based reactive synthesis algorithm ranges from $O((n+1)^2)$ to $O(n^4)$ while the best-case complexity of product automaton-based reactive synthesis algorithms is $O(n^2(n+1)^2)$.

Proof: Suppose the number of atomic propositions of an LTL formula ϕ is n, the cardinality of states in the LOTS is $|S|$, and the set of states of product automaton is Q_p. If the current state is only related to the previous action, the number of automaton states ranges from $n+1$ to n^2. Then, the number of states of the product automaton ranges from $O(|S|(n+1))$ to $O(|S|n^2)$. To fully construct the product automaton graph, the works of [7], [8], [9], and [10] need to check for valid transitions for all pairs $\forall q_p, q_p' \in Q_p$, which results in the complexity of the product automaton-based reactive method ranging from $O(|S|^2(n+1)^2)$ to $O(|S|^2(n^4))$. In contrast, the reactive synthesis in this work searches directly in the automaton $\mathcal{B}$, and the complexity ranges from $O((n+1)^2)$ to $O(n^4)$. By Assumption 1, the given LTL task ϕ can be completed. That is, in the presence of human interference such as dis_1 and dis_2, the number of objects in W is greater than the atomic proposition n. Suppose the state of W at time t is $s_t = o_1^{n_1} o_2^{n_2} o_3^{n_3} \ldots o_x^{n_x}\_o_1^{m_1} o_2^{m_2} o_3^{m_3} \ldots o_x^{m_x}$. then the number of all objects in W is $\sum_{i=1}^{x}(m_i + n_i)$, which indicates that $\sum_{i=1}^{x}(m_i + n_i) \geq n$ and $|S| = \prod_{i=1}^{x}(m_i+1)(n_i+1) > \sum_{i=1}^{x}(m_i + n_i) \geq n$. It can be estimated that the complexity of the product automaton-based reactive synthesis algorithm is at least $O(n^2(n+1)^2)$, while the worst-case of maximum complexity of the LOTS-based approach is $O(n^4)$, which is better than automaton-based reactive approaches. ■

B. Motion Planning Module

The previous section focuses on how the robot determines the actions to be taken based on the locally observed environmental changes. This section presents how to achieve determined actions. For example, consider the action of moving the oranges from W^A into the basket in W^B. To improve the grasping performance, a high-resolution grasping network (HRG-Net) is developed by using the RGB image and depth image of the RGB-D camera in a fully convolutional network, and outputting three feature maps $\mathcal{Q}$, Φ, and $\mathcal{W}$ as shown in Fig. 5. $\mathcal{Q}$ is an image that describes the grasp quality executed at each point $p = (x, y)$. The value q is a scalar that ranges from 0 to 1, where a value closer to 1 indicates higher grasp quality, i.e., the higher chance of grasp success. Φ is an image that describes the grasp angle to be executed at each point p. Since the antipodal grasp is symmetrical around $\pm\frac{\pi}{2}$ radians, the angles are given in the range $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$. $\mathcal{W}$ is an image that describes the gripper width to be executed at each point p. To allow for depth invariance, the variable z is in the range of [0, 150] pixels, which can be converted to a physical measurement using the depth camera parameters and the measured depth. According to the above three feature maps $\mathcal{Q}$, Φ, $\mathcal{W}$ and the object detection algorithm, we can find the best grasping position and pose $g = (x, y, z, \phi, w, q)$.

Although g can be obtained from a single view, such a approach is not preferred, since the target object in general occupies very few pixels at the initial position where the manipulator is often far away from the object (as in Fig. 6(1). A small deviation in prediction can have a great impact on the quality of the grasp. Thus, multiple views method [45] are used to predict the best grasp position and pose g after comprehensive comparisons of different views. Another advantage of the multiple views approach is that it takes full advantage of eye-in-hand features to yield accurate predictions when the camera is close to the object. However, the selection of multiple viewpoints is a difficult problem. Existing methods mainly rely on the gradient of entropy to facilitate the selection of appropriate grasping position and orientation g. Hence, early prediction of the grasping position plays an important role, as poor prediction tends to cause the robotic manipulator to spend more time exploring until it is confident enough to complete the grasping task. If the robotic manipulator reaches the lowest point and still cannot find a suitable grasping point, the grasping task fails. Through experiments shown in Sec. V, we find that traditional CNN methods often fail to get good early grasp predictions.

Unlike traditional CNN based methods, our network does not directly use the encoder and decoder method to obtain the feature map that has the same size as the input depth map. Those traditional CNN based methods are not preferred mainly because, in the process of motion planning, the camera's FOV decreases rapidly due to the eye-in-hand setup, leading to potential failures in searching the target in many scenes. Therefore, there are not enough candidates of predicted grasping locations, resulting in a lot of time spent exploring and still not finding the ideal g. On the contrary, HRG-Net retains the high-resolution representation and convolutes the flow from high resolution to low resolution in parallel, while repeatedly exchanging information. The advantage is that it can both retain accurate spatial information in high resolution, and contain rich semantic information in low resolution.

The visualization of the motion planning module is shown in Fig. 6. In Fig. 6 (a), the robot first generates an initial prediction of the grasping position and pose at the initial position. Then the end-effector moves in the direction that keeps the target in the center of the camera FOV, as indicated by the red arrow in the figure. Fig. 6 (b) and Fig. 6 (c) show that the FOV is gradually reduced as the end-effector moves closer to the target. In Fig. 6 (d), the robot reaches the desired position and is ready to complete the grasping task.

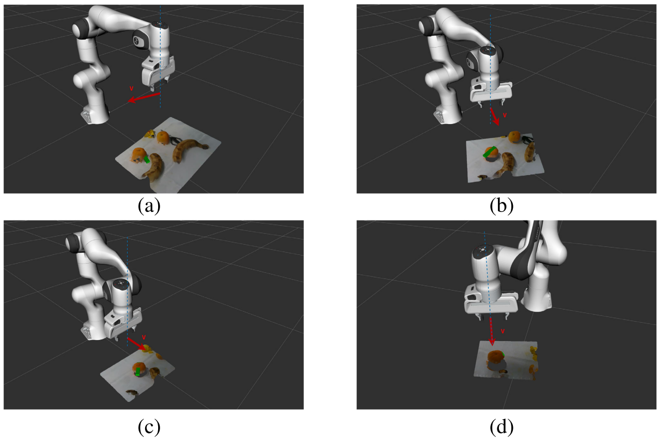

(a) (b) (c) (d)

Fig. 6. These images show that applying HRG-Net in multiple views can improve the performance of the grasping prediction. The objects observed by the camera are shown using a point cloud, and the green rectangular blocks are visualizations of the predicted grasping positions and poses, the red arrow represents the speed direction of the current motion control.

It is worth pointing out that the motion planning of the manipulator only depends on local visual perception. Initially, the manipulator starts from the center of the workspace W and determines the next step direction according to the current local perception of the environment. Particularly, the motion is planned to reduce the uncertainty in the grasping process. For instance, there may be multiple objects in the camera's field of view. To locate the target from these objects, the robot needs to minimize the entropy, so that a high-quality grasp prediction can be obtained by more focusing on the target and reducing possible interference from other objects. Once the manipulator reaches the preset height Z, the robotic manipulator will determine the final predicted gripping posture and position based on the observations throughout the trajectory.

C. Security Mechanisms

We take two types of approaches to address potential safety concerns in human-robot collaboration. The first approach focuses on preventing collisions before they occur (pre-collision measures), while the second approach focuses on limiting the impact force in case of a collision (post-collision measures).

Regarding the pre-collision, we propose the following three points to ensure safety:

- By utilizing the RGB-D camera at the end effector, we can detect the presence of humans in the robot's workspace. If a human is detected and the distance between the end-effector and the human is less than 0.05 m, the robot arm stops to prevent any potential harm. This is an effective way to ensure safety by proactively avoiding collisions.
- We design the framework consisting of a gripping detection network and motion planning algorithm that can effectively avoid the end-effector to enter the singularity, to avoid the situation that the end-effector moving a small distance needs large joint angular velocity. This situation

will seriously cause the robotic arm to lose control, which is the most serious safety hazard.

- We have added various constraints to the program, including speed and acceleration limits. If the robot arm's speed or acceleration exceeds the threshold values, indicating that it is entering a singularity point, the program immediately disconnects control of the robot arm. Additionally, if the robot arm's command frequency drops below 1000 Hz, an emergency stop is triggered to ensure safety. These constraints provide additional safeguards and prevent potentially dangerous situations.

Regarding post-collision, we primarily rely on impedance control enabled by the Franka Panda Emika robot's force sensors, which have a collision detection time of less than $2ms$ and a force-sensing resolution of less than $0.05N$. These sensors enable the robot arm to detect collisions and halt quickly, thereby avoiding any actual harm to humans. Following practical testing, we can confirm that our approach ensures the safety of human-robot collaboration.

V. Experiments

In this section, we first describe the experimental setup and show the performance of the developed HRG-Net on mainstream datasets and physical experiments. We then evaluate the real-time performance of the developed local observation-based reactive planning method via experiments of human-robot collaboration.

The desk-mounted Franka Panda robotic manipulator and the RealSense D435 RGB-D cameras are used in the experiment. The camera is attached to the end-effector of the manipulator.[2] The YOLO3 object detection algorithm [46] is used throughout to detect the type and number of objects relevant to the task at hand. We use a computer with an Ubuntu real-time kernel for robotic manipulation control and another computer equipped with an Nvidia RTX-2080Super for network deployment.

A. HRG-Net Performance

We first validate the performance of HRG-Net on three popular grasping datasets: the Cornell, Jacquard, and Graspnet-1Billion datasets. The results are shown in Table I, Table II, and Table III.

On the Cornell dataset, the image-wise (IW) and object-wise (OW) settings are used. Since the dataset is relatively small, to make the comparison more fair and meaningful, we use five-fold cross-validation following previous works [45], [47]. To show the computational time of HRG-Net, the average speed of processing a single image using NVIDIA RTX-2080 Super is about 51ms, which meets the requirement of real-time implementation. On the Jacquard dataset, we split the entire dataset into a 90% training set and a 10% test set. We tested the performance of HRG-Net in the RGB, depth, and RGB-D channels separately. We also tested HRG-Net on the Graspnet-1billion dataset which has 97280 RGB images and each image is captured from multiple views in over 190 cluttered scenarios with real-world sensors. HRG-Net achieves better performance in all input modalities compared with previous methods. Our method achieves state-of-the-art in those datasets. These results also indicate that better accuracy can be achieved by combining depth image and RGB image, which is consistent with common sense, since fusing multi-modal information like color and depth images enables information crossover and complement, thus improving the robustness and generalization of the network. The code is released for validation.[3]

TABLE I

The Accuracy on Cornell Grasping Dataset

Method	Input	Accuracy(%)		Time (ms)
		IW	OW	
Fast Search [48]	RGB-D	60.5	58.3	5000
GG-CNN [49]	D	73.0	69.0	**19**
SAE [36]	RGB-D	73.9	75.6	1350
Two-stage closed-loop [40]	RGB-D	85.3	-	140
AlexNet, MultiGrasp [50]	RGB-D	88.0	87.1	76
STEM-CaRFs [51]	RGB-D	88.2	87.5	-
GRPN [52]	RGB	88.7	-	200
ResNet-50x2 [53]	RGB-D	89.2	88.9	103
GraspNet [54]	RGB-D	90.2	90.6	24
ZF-net [55]	RGB-D	93.2	89.1	-
E2E-net [56]	RGB	98.2	-	63
GR-ConvNet [57]	D	93.2	94.3	**19**
	RGB	96.6	95.5	**19**
	RGB-D	97.7	96.6	20
TF-Grasp [39]	D	95.2	94.9	41.1
	RGB	96.78	95.0	41.3
	RGB-D	97.99	96.7	41.6
HRG-Net	D	**99.43**	**96.8**	52.6
	RGB	**98.50**	**96.7**	53.0
	RGB-D	**99.50**	**97.5**	53.7

TABLE II

The Accuracy on Jacquard Grasping Dataset

Authors	Method	Input	Accuracy (%)
Depierre [37]	Jacquard	RGB-D	74.2
Morrison [49]	GG-CNN2	D	84
Zhou [38]	FCGN, ResNet-101	RGB	91.8
Alexandre [58]	GQ-STN	D	70.8
Zhang [59]	ROI-GD	RGB	90.4
Stefan [56]	Det Seg	RGB	92.59
Stefan [56]	Det Seg Refine	RGB	92.95
Kumra [57]	GR-ConvNet	D	93.7
Kumra [57]	GR-ConvNet	RGB	91.8
Kumra [57]	GR-ConvNet	RGB-D	94.6
Wang [39]	TF-Grasp	D	93.1
Wang [39]	TF-Grasp	RGB	93.57
Wang [39]	TF-Grasp	RGB-D	94.6
Our	HRG-Net	D	**95.8**
	HRG-Net	RGB	**95.7**
	HRG-Net	RGB-D	**96.5**

TABLE III

The Accuracy on GraspNet-1Billion Results

Input	Training	Seen	Unseen	Novel
RGB	93%	89%	70%	73%
Depth	94%	87%	77%	73%
RGB-D	96%	90%	82%	79%

[2] The eye-in-hand manipulation is preferred in this work since it generally enables precise control as images can be captured much closer to an object.

[3] https://github.com/USTCzzl/HRG_Net

TABLE IV
COMPARISON RESULTS USING AND NOT USING LAYER-FUSION

The accuracy on Cornell Grasping Results		
	With Layer-fusion	Without Layer-fusion
RGB	98.50%	95.0%
Depth	99.43%	96.0%
RGB+Depth	99.50%	98.50%
The accuracy on Jacquard Grasping Results		
	With Layer-fusion	Without Layer-fusion
RGB	96.7%	92.4%
Depth	96.8%	91.8%
RGB+Depth	97.5%	93.27%

To investigate the role of layer-fusion in HRG-Net, we did ablation studies on Cornell and Jacquard datasets with and without layer-fusion, respectively. The experimental results are shown in Table IV. The use of layer-fusion is significantly better than not using layer-fusion in both datasets. This is because the information exchange that occurs between feature maps of different sizes allows the network to take into account both low-level spatial features and high-level semantic features.

Combined with the object detection algorithm, HRG-Net can locate specific objects in the clutter and generate grasp bounding box predictions. Since the visual processing time is negligible during the whole robotic manipulation grasping process, we sample multiple viewpoints during the robotic manipulation motion and then select the best grasping position and orientation from them. Since the camera is far away from the object at the beginning, the pixels of each object are relatively low and thus the capture position is not ideal. As the camera gets closer to the object, this issue can be gradually avoided. For instance, as shown in Fig. 6 (a) If the robotic manipulator directly follows the prediction at this point to grasp orange can easily lead to failure, since the predicted grasp position is close to the banana at the beginning. However, as the camera approaches the orange, the predicted position and orientation of the grasp box are significantly improved in Fig. 6 (b)-(d)). During this process, the robot will call HRG-Net many times until a preset height is reached (0.2m in the experiment). We sampled several points on the motion trajectory of the robotic manipulator, and, as the view is getting smaller and smaller, only part of the W^A can be perceived, which motivates the introduction of LOTS.

To further test the robustness of HRG-Net using local observation, we designed three sets of comparison experiments, namely, grasping under multi-object interference, grasping under random initial positions, and grasping with dynamic targets. In each set of experiments, HRG-Net is compared with the methods of [45] and [47] in terms of the success rate and the time consumption. The experimental results are shown in Fig. 7, which indicate that HRG-Net is not only highly accurate but also takes the least amount of time. This is because HRG-Net retains high-resolution features throughout the network and is, therefore, able to make good predictions in the preparation phase of the capture, reducing the time spent exploring for ideal position and orientation. For more details, please refer to the video https://www.youtube.com/watch?v=nJsql6b3W1Q.

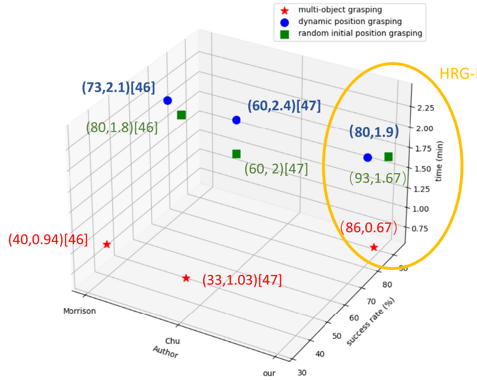


Fig. 7. The performance comparison of HRG-Net with [45] and [47]. The x-axis indicates the success rate and the z-axis indicates the time consumption used to complete the task.

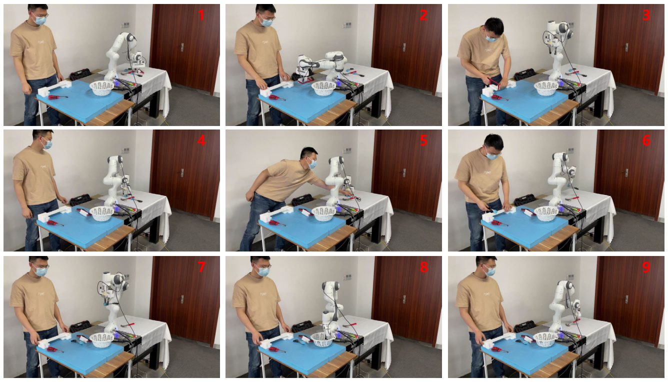


Fig. 8. The Human operator and robotic manipulator work in a shared workspace. Fig. 8.5-Fig. 8.9 demonstrate that, when the human is assistive, the robot can determine the current task progress and continue on subsequent tasks. The experiment video is available at https://youtu.be/jKXuOJu1MsA.

To investigate the role of layer-fusion in HRG-Net, we did ablation studies on Cornell and Jacquard datasets with and without layer-fusion, respectively. The experimental results are shown in Table IV. The use of layer-fusion is significantly better than not using layer-fusion in both datasets. This is because the information exchange that occurs between feature maps of different sizes allows the network to take into account both low-level spatial features and high-level semantic features.

B. Experiment of Human-Robot Collaboration in Intelligent Manufacturing

1) Model Building: The manufacturing task is specified as $\phi_{model} = \Box\Diamond p1 \wedge \Box\Diamond p2 \wedge \Box\Diamond p3$ (The order is not reflected in the formula because we used the default order of the LTL2BA tool[4]), where $p1$ indicates the action of taking the saws from W^A to W^B, $p2$ indicate the actions of taking the scissors from W^A to W^B, and $p3$ indicates the action of taking the hammer from W^A to W^B. The snapshots of the experiment are presented in Fig. 8. Initially, in Fig. 8.1 - Fig. 8.3, the robot picks up the saw from W^A and places it onto W^B, after which the human operator starts using the saw to cut objects. In Fig. 8.4 - Fig. 8.6, the human operator desires to remove burrs from the objects as early as possible and therefore assists the robot by moving the scissors from W^A to

[4] https://github.com/utwente-fmt/ltl2ba

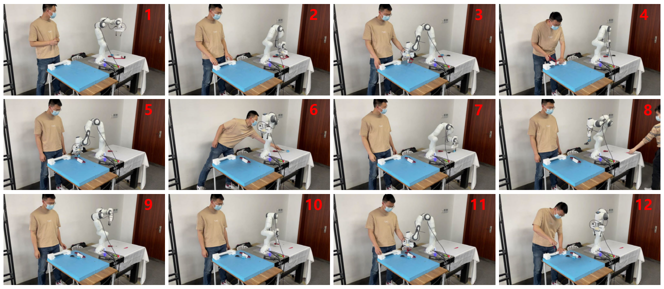


Fig. 9. The Human operator and robotic manipulator work in a shared workspace. Fig. 9.6-Fig. 9.12 demonstrate that, when the humans are non-assistive, the robot can react promptly to ensure mission completion. The experiment video is available at https://youtu.be/jKXuOJu1MsA.

W^B and quickly starts removing the burrs using the scissors. This corresponds to dis_1 as defined in Sec. III-B, since the type and number of objects in the workspace W remain unchanged. However, this process is not captured by the local sensing of the robot arm. As a result, the robot arm continues to be ready to transport the scissors from W^A to W^B according to the task formula ϕ_{model}, until it eventually realizes that the subtask has already been completed. Once the robot arm recognizes this, it places the scissors in the waste basket, which can be seen in Fig. 8.7 - Fig. 8.8. Finally, in Fig. 8.9, the robot arm picks up the hammer from W^A and transports it to W^B, thereby completing the task as defined by ϕ_{model}. The robotic arm was able to successfully handle non-assisted human operations as well. For instance, in Fig. 9.5 - Fig. 9.6, the robotic arm completed the subtask of sending scissors from W^A to W^B, but the human mistakenly sent the scissors back to W^A. The robotic arm recognized this interference and retrieved the scissors again from W^A to W^B. This example falls under dis_1 in Sec. III-B since it did not change the type or number of objects in the workspace W. In contrast, Fig. 9.8 is an example of dis_2 in Sec. III-B, where the human introduced a new tool that changed the type and number of objects in workspace W, but this did not interfere with the robot's decision to continue performing subtask $p3$.

2) Human-Machine Collaboration to Build Towers: To further demonstrate the effectiveness of the developed reactive planning framework, human-robot collaboration in building a toy tower is considered in this section. Building a tower is order-sensitive, i.e., the towering peak, the tower body, and the tower bottom have to be stacked in a specific order. In the experiment, the blocks on W^A are required to be moved to W^B and stacked in order to build a tower. Such a task is specified in the LTL formula as $\phi_{stacking} = \Diamond(p1\bigcirc(\Diamond(p2\bigcirc(\Diamond p3))))$, where $p1$, $p2$, and $p3$ indicates the action of moving the tower bottom, tower body, and tower peak form W^A to W^B, respectively. The task $\phi_{stacking}$ requires first moving the tower bottom from W^A to W^B, placing the tower body on top of the tower bottom, and finally placing the towering peak on top of the tower body. The snapshots of the experiment are shown in Fig. 10. Fig. 10.1 - Fig. 10.3 shows the robot arm first completes the subtask $p1$. Fig. 10.4 shows the human interference of type dis_2 (introducing a new object to the W^B). Human has finished subtask $p2$. Since the robotic arm only has local perception and does not find the subtask $p2$ being completed, the robotic arm will still look for the tower body. Fig. 10.5 - Fig. 10.6: when the robot arm is ready to place the tower body, it found that $p2$ had been finished, so it put the tower aside and went to look for the top of the tower in W^A. Fig. 10.7 - Fig. 10.8: when the robot arm was looking for the towering peak, the human mistakenly took the tower body away from W^B (interference of type dis_2). When the robot arm was ready to place the towering peak, it found that $p2$ has not been completed, so it put the towering peak back. Fig. 10.9 - Fig. 10.13: when the robot arm rebuilt the tower body, the human moved the tower body from W^B to W^A by mistake (interference of type dis_1). The robot arm then went back to find the tower body to rebuild the tower body. At this point, the robot arm completes subtask $p2$. The human interference is of type dis_2 (i.e., introducing a new object to the W^A). Since such interference does not affect the desired order of building the tower, the robot decides to continue to select the towering peak and place it on top of the tower body, as shown in Fig. 10.14 - Fig. 10.16. Subtask $p3$ ends and the whole task is completed.

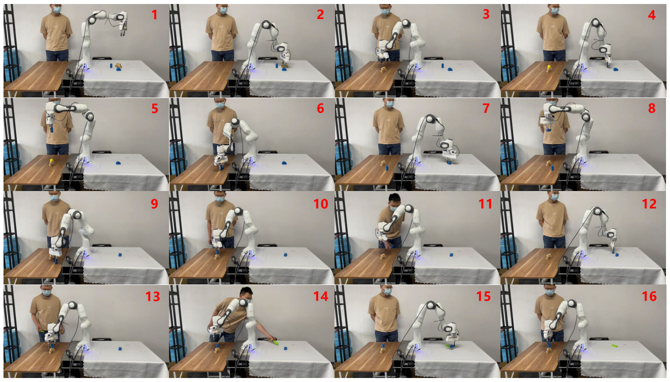


Fig. 10. In the human-machine collaborative tower building experiment, the robotic arm needs to observe human behavior to passively take action to ensure the tower is built successfully. The experiment video is available at https://youtu.be/yUs-abJtbuc.

VI. Conclusion

This study presents a real-time reactive planning framework that enables a robot to perceive its surroundings and infer the progress of the current task, as well as the impact of human collaborators, using only the camera in its hand. The framework allows the robot to quickly react and reprogram its actions during human-robot collaboration. To ensure the real-time robustness of this planning framework, we have developed a new grasping location prediction network called HRG-Net. HRG-Net achieves promising performance compared to existing methods. We have also designed a closed-loop grasping strategy that effectively avoids robots from entering singularities and ensures the safety of human-robot collaboration.

Our ongoing research will consider using timed temporal logic to impose time constraints on reactive planning and further extend the application scenarios of existing methods. Additionally, we will investigate the prediction of the behavior of human collaborators. Predicting human behavior allows for faster reactive planning and improved efficiency, and can be combined with control methods such as control barrier functions to ensure the safety of human-machine collaboration.

REFERENCES

[1] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," *Auto. Robots*, vol. 42, no. 5, pp. 957–975, Jun. 2018.

[2] Z. Li, G. Li, X. Wu, Z. Kan, H. Su, and Y. Liu, "Asymmetric cooperation control of dual-arm exoskeletons using human collaborative manipulation models," *IEEE Trans. Cybern.*, vol. 52, no. 11, pp. 12126–12139, Nov. 2022.

[3] T. Liu, E. Lyu, J. Wang, and M. Q.-H. Meng, "Unified intention inference and learning for human–robot cooperative assembly," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 3, pp. 2256–2266, Jul. 2022.

[4] B. Huang, Y. Yang, Y.-Y. Tsai, and G.-Z. Yang, "A reconfigurable multirobot cooperation workcell for personalized manufacturing," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 3, pp. 2581–2590, Jul. 2022.

[5] X. Wu and Z. Li, "Cooperative manipulation of wearable dual-arm exoskeletons using force communication between partners," *IEEE Trans. Ind. Electron.*, vol. 67, no. 8, pp. 6629–6638, Aug. 2020.

[6] G. Li, Z. Li, and Z. Kan, "Assimilation control of a robotic exoskeleton for physical human–robot interaction," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 2977–2984, Apr. 2022.

[7] S. Li, D. Park, Y. Sung, J. A. Shah, and N. Roy, "Reactive task and motion planning under temporal logic specifications," 2021, *arXiv:2103.14464*.

[8] K. He, A. M. Wells, L. E. Kavraki, and M. Y. Vardi, "Efficient symbolic reactive synthesis for finite-horizon tasks," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 8993–8999.

[9] K. He, M. Lahijanian, L. E. Kavraki, and M. Y. Vardi, "Reactive synthesis for finite tasks under resource constraints," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 5326–5332.

[10] M. Wells, Z. Kingston, M. Lahijanian, L. E. Kavraki, and M. Y. Vardi, "Finite-horizon synthesis for probabilistic manipulation domains," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 6336–6342, doi: 10.1109/ICRA48506.2021.9561297.

[11] Z. Zhou, S. Wang, Z. Chen, M. Cai, and Z. Kan, "A robotic visual grasping design: Rethinking convolution neural network with high-resolutions," 2022, *arXiv:2209.07459*.

[12] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "GraspNet-1Billion: A large-scale benchmark for general object grasping," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11441–11450.

[13] C. Baier and J.-P. Katoen, *Principles of Model Checking*. Cambridge, MA, USA: MIT Press, 2008.

[14] W. Wang, R. Li, Y. Chen, Y. Sun, and Y. Jia, "Predicting human intentions in human–robot hand-over tasks through multimodal learning," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 3, pp. 2339–2353, Jul. 2022.

[15] M. Raessa, J. C. Y. Chen, W. Wan, and K. Harada, "Human-in-the-loop robotic manipulation planning for collaborative assembly," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 4, pp. 1800–1813, Oct. 2020.

[16] H. Bai, D. Hsu, and W. S. Lee, "Integrated perception and planning in the continuous space: A POMDP approach," *Int. J. Robot. Res.*, vol. 33, no. 9, pp. 1288–1302, Aug. 2014.

[17] S. Ghosh and J. Biswas, "Joint perception and planning for efficient obstacle avoidance using stereo vision," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1026–1031.

[18] E. Seraj, L. Chen, and M. C. Gombolay, "A hierarchical coordination framework for joint perception-action tasks in composite robot teams," *IEEE Trans. Robot.*, vol. 38, no. 1, pp. 139–158, Feb. 2022.

[19] M. Ghasemi, E. Bulgur, and U. Topcu, "Task-oriented active perception and planning in environments with partially known semantics," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 3484–3493.

[20] J. Tumova, S. Karaman, C. Belta, and D. Rus, "Least-violating planning in road networks from temporal logic specifications," in *Proc. ACM/IEEE 7th Int. Conf. Cyber-Phys. Syst. (ICCPS)*, Apr. 2016, pp. 1–9.

[21] C.-I. Vasile, J. Tumova, S. Karaman, C. Belta, and D. Rus, "Minimum-violation scLTL motion planning for mobility-on-demand," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 1481–1488.

[22] M. Cai, S. Xiao, Z. Li, and Z. Kan, "Optimal probabilistic motion planning with potential infeasible LTL constraints," *IEEE Trans. Autom. Control*, vol. 68, no. 1, pp. 301–316, Jan. 2023.

[23] Z. Li, M. Cai, S. Xiao, and Z. Kan, "Online motion planning with soft metric interval temporal logic in unknown dynamic environment," *IEEE Control Syst. Lett.*, vol. 6, pp. 2293–2298, 2022.

[24] M. Guo and D. V. Dimarogonas, "Multi-agent plan reconfiguration under local LTL specifications," *Int. J. Robot. Res.*, vol. 34, no. 2, pp. 218–235, Feb. 2015.

[25] S. Andersson and D. V. Dimarogonas, "Human in the loop least violating robot control synthesis under metric interval temporal logic specifications," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2018, pp. 453–458.

[26] S. Ahlberg and D. V. Dimarogonas, "Human-in-the-loop control synthesis for multi-agent systems under hard and soft metric interval temporal logic specifications," in *Proc. IEEE 15th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2019, pp. 788–793.

[27] Q. Lu and Q.-L. Han, "Mobile robot networks for environmental monitoring: A cooperative receding horizon temporal logic control approach," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 698–711, Feb. 2019.

[28] M. Cai, H. Peng, Z. Li, H. Gao, and Z. Kan, "Receding horizon control based motion planning with partially infeasible LTL constrains," *IEEE Control Syst. Lett.*, vol. 5, no. 4, pp. 1279–1284, Apr. 2020.

[29] M. Cai, H. Peng, Z. Li, and Z. Kan, "Learning-based probabilistic LTL motion planning with environment and motion uncertainties," *IEEE Trans. Autom. Control*, vol. 66, no. 5, pp. 2386–2392, May 2021.

[30] X. Li, Z. Serlin, G. Yang, and C. Belta, "A formal methods approach to interpretable reinforcement learning for robotic planning," *Sci. Robot.*, vol. 4, no. 37, pp. 1–12, Dec. 2019.

[31] B. Lacerda, F. Faruq, D. Parker, and N. Hawes, "Probabilistic planning with formal performance guarantees for mobile service robots," *Int. J. Robot. Res.*, vol. 38, no. 9, pp. 1098–1123, Aug. 2019.

[32] C. I. Vasile, X. Li, and C. Belta, "Reactive sampling-based path planning with temporal logic specifications," *Int. J. Robot. Res.*, vol. 10, Jan. 2020, Art. no. 0278364920918919.

[33] V. Vasilopoulos, Y. Kantaros, G. J. Pappas, and D. E. Koditschek, "Reactive planning for mobile manipulation tasks in unexplored semantic environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 6385–6392.

[34] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proc. Millennium Conf., IEEE Int. Conf. Robot. Automat., Symposia*, San Francisco, CA, USA, Apr. 2000, pp. 348–353.

[35] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL, USA: CRC Press, 1994, 2017.

[36] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *Int. J. Robot. Res.*, vol. 34, nos. 4–5, pp. 705–724, Apr. 2015.

[37] A. Depierre, E. Dellandréa, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 3511–3516.

[38] X. Zhou, X. Lan, H. Zhang, Z. Tian, Y. Zhang, and N. Zheng, "Fully convolutional grasp detection network with oriented anchor box," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 7223–7230.

[39] S. Wang, Z. Zhou, and Z. Kan, "When transformer meets robotic grasping: Exploits context for efficient grasp detection," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 8170–8177, Jul. 2022.

[40] Z. Wang, Z. Li, B. Wang, and H. Liu, "Robot grasp detection using multimodal deep convolutional neural networks," *Adv. Mech. Eng.*, vol. 8, no. 9, 2016, Art. no. 1687814016668077.

[41] J. Song, M. Patel, and M. Ghaffari, "Fusing convolutional neural network and geometric constraint for image-based indoor localization," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1674–1681, Apr. 2022.

[42] Q. Yu, W. Shang, Z. Zhao, S. Cong, and Z. Li, "Robotic grasping of unknown objects using novel multilevel convolutional neural networks: From parallel gripper to dexterous hand," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 4, pp. 1730–1741, Oct. 2021.

[43] E. M. Clarke Jr., O. Grumberg, D. Kroening, D. Peled, and H. Veith, *Model checking*. Cambridge, MA, USA: MIT Press, 2018.

[44] Y. Kantaros and M. M. Zavlanos, "Distributed optimal control synthesis for multi-robot systems under global temporal tasks," in *Proc. ACM/IEEE 9th Int. Conf. Cyber-Phys. Syst. (ICCPS)*, Apr. 2018, pp. 162–173.

[45] D. Morrison, P. Corke, and J. Leitner, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," 2018, *arXiv:1804.05172*.

[46] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[47] F.-J. Chu, R. Xu, and P. A. Vela, "Real-world multiobject, multigrasp detection," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3355–3362, Oct. 2018.

[48] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from RGBD images: Learning using a new rectangle representation," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 3304–3311.

[49] D. Morrison, P. Corke, and J. Leitner, "Learning robust, real-time, reactive robotic grasping," *Int. J. Robot. Res.*, vol. 39, nos. 2–3, pp. 183–201, Mar. 2020.

[50] J. Redmon and A. Angelova, “Real-time grasp detection using convolutional neural networks,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 1316–1322.

[51] U. Asif, M. Bennamoun, and F. A. Sohel, “RGB-D object recognition and grasp detection using hierarchical cascaded forests,” *IEEE Trans. Robot.*, vol. 33, no. 3, pp. 547–564, Jun. 2017.

[52] H. Karaoguz and P. Jensfelt, “Object detection approach for robot grasp detection,” in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 4953–4959.

[53] S. Kumra and C. Kanan, “Robotic grasp detection using deep convolutional neural networks,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 769–776.

[54] U. Asif, J. Tang, and S. Harrer, “GraspNet: An efficient convolutional neural network for real-time grasp detection for low-powered devices,” in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 4875–4882.

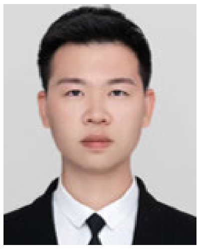
[55] D. Guo, F. Sun, H. Liu, T. Kong, B. Fang, and N. Xi, “A hybrid deep architecture for robotic grasp detection,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 1609–1614.

[56] S. Ainetter and F. Fraundorfer, “End-to-end trainable deep neural network for robotic grasp detection and semantic segmentation from RGB,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 13452–13458.

[57] S. Kumra, S. Joshi, and F. Sahin, “Antipodal robotic grasping using generative residual convolutional neural network,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 9626–9633.

[58] A. Gariépy, J.-C. Ruel, B. Chaib-draa, and P. Giguère, “GQ-STN: Optimizing one-shot grasp detection based on robustness classifier,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 3996–4003.

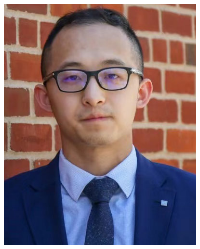
[59] H. Zhang, X. Lan, S. Bai, X. Zhou, Z. Tian, and N. Zheng, “ROI-based robotic grasp detection for object overlapping scenes,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 4768–4775.



Zhangli Zhou received the B.E. degree from the Wuhan University of Technology, Wuhan, China, in 2016. He is currently pursuing the Ph.D. degree with the University of Science and Technology of China, Hefei, China. His research interests include robotics and reactive task and motion planning.

Shaochen Wang (Graduate Student Member, IEEE) received the B.E. degree from the Hefei University of Technology, Hefei, China, in 2016. He is currently pursuing the Ph.D. degree in control science and engineering with the University of Science and Technology of China, Hefei. His current research interests include reinforcement learning and robotics.

Ziyang Chen received the B.E. degree from Beihang University, Beijing, China, in 2020. He is currently pursuing the Ph.D. degree in control science and engineering with the University of Science and Technology of China, Hefei, China. His current research interests include robotics and multi-agent systems.



Mingyu Cai received the Ph.D. degree in mechanical engineering from the University of Iowa, Iowa City, IA, USA, in 2021. From 2021 to 2023, he was a Post-Doctoral Associate with the Department of Mechanical Engineering, Lehigh University, Bethlehem, PA, USA. He was a Research Scientist with the Honda Research Institute. Since 2024, he has been an Assistant Professor with the University of California at Riverside, Riverside, CA, USA. His research interests include robotics, machine learning, control theory, and formal methods, with a focus on applications in motion planning, decision-making, nonlinear control, and autonomous driving.

Hao Wang received the B.S. degree in mechanical engineering from the China University of Mining and Technology, Xuzhou, Jiangsu, China, in 2017. He is currently pursuing the Ph.D. degree in automation with the University of Science and Technology of China, Hefei, China. His current research interests include manipulation skill learning and deep reinforcement learning.

Zhijun Li (Fellow, IEEE) received the Ph.D. degree in mechatronics from Shanghai Jiao Tong University in 2002. From 2003 to 2005, he was a Post-Doctoral Fellow with the Department of Mechanical Engineering and Intelligent Systems, The University of Electro-Communications, Japan. From 2005 to 2006, he was a Research Fellow with the Department of Electrical and Computer Engineering, National University of Singapore, and Nanyang Technological University, Singapore. Since 2017, he has been a Professor with the Department of Automation, University of Science and Technology of China, where he has been the Vice Dean of the School of Information Science and Technology since 2019. His current research interests include wearable robotics, bio-mechatronics systems, nonlinear control, and computational optimization. He is a fellow of AAIA. He is a member of the Board of Governors and the IEEE Systems, Man and Cybernetics Society (2023–2025). He has been the Co-Chair of the IEEE SMC Technical Committee on Bio-Mechatronics and Bio-Robotics Systems and the IEEE RAS Technical Committee on Neuro-Robotics Systems. He is an associate editor of several IEEE TRANSACTIONS.

Zhen Kan (Senior Member, IEEE) received the Ph.D. degree from the Department of Mechanical and Aerospace Engineering, University of Florida, in 2011. He was a Post-Doctoral Research Fellow with the Air Force Research Laboratory (AFRL), Eglin AFB, and the University of Florida REEF from 2012 to 2016. He was an Assistant Professor with the Department of Mechanical Engineering, University of Iowa, from 2016 to 2019. He is currently a Professor with the Department of Automation, University of Science and Technology of China. His research interests include networked control systems, nonlinear control, formal methods, and robotics. He also serves on the program committees for several internationally recognized scientific and engineering conferences. He is an Associate Editor of IEEE TRANSACTIONS ON AUTOMATIC CONTROL and IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.