Human-Aware Reactive Task Planning of Sequential Robotic Manipulation Tasks

Wanyu Ma, *Member, IEEE*, Anqing Duan, *Member, IEEE*, Hoi-Yin Lee, *Student Member, IEEE*, Pai Zheng, *Senior Member, IEEE*, and David Navarro-Alarcon, *Senior Member, IEEE*

***Abstract*—The recent emergence of Industry 5.0 underscores the need for increased autonomy in human–robot interaction (HRI), presenting both motivation and challenges in achieving resilient and energy-efficient production systems. To address this, in this article, we introduce a strategy for seamless collaboration between humans and robots in manufacturing and maintenance tasks. Our method enables smooth switching between temporary HRI (human-aware mode) and long-horizon automated manufacturing (fully automatic mode), effectively solving the human–robot coexistence problem. We develop a task progress monitor that decomposes complex tasks into robot-centric action sequences, further divided into three-phase subtasks. A trigger signal orchestrates mode switches based on detected human actions and their contribution to the task. In addition, we introduce a human agent coefficient matrix, computed using selected environmental features, to determine cut-points for reactive execution by each robot. To validate our approach, we conducted extensive experiments involving robotic manipulators performing representative manufacturing tasks in collaboration with humans. The results show promise for advancing HRI, offering pathways to enhancing sustainability within Industry 5.0. Our work lays the foundation for intelligent manufacturing processes in future societies, marking a pivotal step toward realizing the full potential of human–robot collaboration.**

***Index Terms*—Autonomy level (AL), human–robot interaction (HRI), reactive task planning, sequential robotic manipulation.**

Notation

Bold capital, bold small, and italic small letters denote matrices, column vectors, and scalars, respectively. $\wedge$, $\vee$, and $\neg$ represent the AND, OR, and NOT Boolean operators, respectively.

n	Number of human–robot agents in the system.
N_k	Number of subtasks for the kth robot.
i, j, k	Indices of the subtasks, phases, and agents.
$\mathcal{S}_k^i$	ith subtask for the kth robot.
$\mathcal{A}_k$, $\mathbf{x}_k$	kth agent, and its pose.
$\mathbf{y}_k^i$	Home pose of the kth robot when $\mathcal{S}_k^i$ ends.
$\mathcal{O}_k^i$, $\mathbf{o}_k^i$, $\mathbf{o}_k^{i*}$	Object manipulated by the kth robot during $\mathcal{S}_k^i$, its pose, and its target pose.
a_k^i	Active coefficient of $\mathcal{S}_k^i$ for the kth robot.
$\mathbf{r}_k^i$, $\bar{\mathbf{r}}_k^i$	Phases' feature vectors for the kth agent during $\mathcal{S}_k^i$. The former uses the information related to the kth agent and the latter uses the information related to the other agents.
$\mathbf{p}_k^i$, $\hat{\mathbf{p}}_k^i$	Phase priority of the kth agent during the ith subtask under AM and HM, respectively.
$\mathbf{m}_k$, $\hat{\mathbf{m}}_k$	Task progress monitor of the kth agent under AM and HM, respectively.
h	Automatic/HM trigger signal.
d_k^{ij}	Geometric features to describe the phases.
$\mathbf{f}$	Environmental feature vector of the reactive tasks.
$\mathbf{\Omega}_H$	Human agent coefficient matrix.

Received 21 May 2024; revised 1 October 2024 and 16 November 2024; accepted 29 November 2024. This work was supported by the Research Grants Council (RGC) of Hong Kong under Grant C4042-23GF. Paper no. TII-24-2473. *(Corresponding author: David Navarro-Alarcon.)*

Wanyu Ma is with the Chinese University of Hong Kong (CUHK), Hong Kong (e-mail: wyma@surgery.cuhk.edu.hk).

Anqing Duan is with the Mohamed Bin Zayed University of Artificial Intelligence, Masdar, UAE (e-mail: anqing.duan@mbzuai.ac.ae).

Hoi-Yin Lee, Pai Zheng, and David Navarro-Alarcon are with The Hong Kong Polytechnic University, Hong Kong (e-mail: hoi-yin.lee@connect.polyu.hk; pai.zheng@polyu.edu.hk; dna@ieee.org).

This article has supplementary downloadable material available at https://doi.org/10.1109/TII.2024.3514130, provided by the authors.

Digital Object Identifier 10.1109/TII.2024.3514130

I. Introduction

In the recently proposed Industry 5.0 paradigm, as highlighted by Li et al. [1], the emphasis is placed on achieving seamless collaboration between humans and robots to jointly perform manufacturing processes. This paradigm embodies the concept of full automation with short-term human–robot interaction (HRI), applicable in scenarios where both robots and human workers contribute their respective skills and strengths, such as collaboratively assembling components, or where robots must safely respond to humans operating in close proximity, such as during maintenance tasks.

However, to realize this level of human–robot coexistence, several technical challenges must be addressed. First, the system must effectively monitor the progress of manufacturing tasks amidst the unpredictable behavior of human agents (the typical changes on task progress after human-aware mode (HM) can be represented with the structure shown in Fig. 1). Second, it should operate in an HM, detecting and responding to human presence or passive contributions to the task. Finally, the system should autonomously transition between automatic mode (AM) and HM to fulfill expected functions based on the sensed environmental state. Despite recent advancements in HRI and intelligent manufacturing, achieving these advanced capabilities remains an open research problem in the literature. This article aims to address these challenges.

Numerous studies on HRI have explored various aspects such as perception [2], [3], communication [4], role assignment [5], [6], task allocation [7], [8], etc. A common factor in these studies is the continuous collaboration that combines the dexterity and

1941-0050 © 2024 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

IEEE Proof

Fig. 1. Typical changes on task progress after HM. (a) Fully AM. (b) No changes on task progress. (c) HM advances the task progress. (d) Past subtask is activated. (e) System needs to complete the interrupted subtask properly, and then, switches to the newly active subtask.

intelligence of human workers with the stability and endurance of automated machines. However, a more comprehensive approach to practical manufacturing processes involves developing methods that allow a fully automated robotic system to temporarily interact with humans. In such an HRI paradigm, a human worker may enter the production area and/or collaboratively contribute to the task [9], [10]. After this interim coexistence scenario, a swift automatic operation resumption can enhance the robustness and environmental friendliness of the manufacturing system. Motivated by these frontier industrial requirements, this article investigates efficient robot control strategies to smoothly switch between a fully AM and HM while continuing the sharing operation [11]. To switch from AM to HM, we adopt a safety-rated monitored operation stop in accordance with the ISO/TS 15066 standard guidelines. In this scenario, the robot halts if a human enters the collaborative workspace and resumes its tasks when the human leaves it [9], [12], [13]. Therefore, our study focuses on strategies to switch from HM to AM.

HRI can be classified into three categories: human–robot coexistence, cooperation, and collaboration. These classifications are based on factors such as space, contact, task, and context awareness [14]. The process of transitioning between these HRI types is known as autonomy level (AL) adjustment [15]. In this framework, an increase in the robotic system's AL correlates with a decrease in the necessity for HRI. Various strategies have been devised to accommodate different ALs, such as evaluating the effectiveness of human–robot teams to determine the appropriate AL [16], adapting policies for joint dyadic tasks [17], adjusting AL for remote vehicle control [18], managing multiple interactions among humans, robots, and their environment [19], and optimizing production cycles by allowing robots to undertake alternative tasks while human operators are engaged in the process [20]. Despite these advancements, existing decision-making and trajectory-planning methods often overlook the nuances of switching from human manipulation (HM) to automated manipulation (AM) and lack the ability required for varied sequential tasks. Regarding the switch between manual mode and AM, there have been different methods to deal with it. For example, Li et al. [21] used temporal logic specifications to synthesize the switching. Fusaro et al. [22] leveraged behavior trees to coordinate humans and robots to complete a task. Braun et al. [23] developed an online robot execution time estimation approach to increase the efficacy of the human-aware controller. By contrast, our switching strategy is able to deal with long sequential tasks, which makes it more suitable for a complex manufacturing scenarios.

To implement AL adjustment, Rabby et al. [15] proposed a method based on reinforcement learning that allows for automatic AL decrease when the system needs to adapt to workspaces of varying sizes or changes in the human operator's rewards. However, this method is tailored to a specific task (placing an object in a container), limiting its applicability to more complex tasks, such as those of a sequential nature. Role assignment is another strategy related to AL adjustment. For example, some approaches design two separate motion planning modules for leader–follower robots [6], while others utilize a weighting function to enable a continuous transition between these behaviors [5]. Nevertheless, most existing methods are designed for specific tasks and cannot be directly applied to the HM/AM transitioning scenario addressed in this article, where a sequential manipulation task must be jointly performed by a human–robot team [24].

Resuming the operation of long-horizon tasks in AM is more complex than merely restarting a machine after a human intervention. This complexity arises from the sequential nature of such tasks, where the context and subtasks determine the overall progress. Unlike simple machine restarts, resuming a long-horizon task with HM requires analyzing the phases constituting the process. Achieving this necessitates developing an intelligent method to monitor task progress and identify the appropriate cut-point for effectively switching from HM to AM. Many works have utilized task scheduling or allocation to replan these types of sequential tasks. For instance, Notomista et al. [25] proposed an algorithm to allocate tasks to robots, taking energy consumption into account. Similarly, Yu et al. [26] drew parallels between selecting chess piece moves and decision making in human–robot collaboration assembly, while Flacco [27] introduced a task priority matrix to establish the correct priority order. In addition, Hong et al. [28] designed a task-priority-based controller for a rescue robot applications. However, these methods are not sufficiently general to provide cut-points for AL adjustment in sequential manipulation tasks conducted by single/multiple robotic systems.

As a feasible solution to the aforementioned issues, this article introduces a novel reactive task-planning method, making the following original contributions.

1) We develop a general structure that decomposes and schedules sequential robotic manipulation tasks taking into account the impact of both human agents and other robots involved in the task.
2) We develop an automatic/human-aware trigger algorithm capable of detecting human motion and assessing its contribution to task progress.
3) We conduct a comprehensive experimental study to validate the proposed method using representative sequential manipulation tasks.

II. Problem Formulation

Definition 1 (Substask $\mathcal{S}_k^i$): In this work, a manipulation subtask $\mathcal{S}_k^i$ represents a *pick-and-place*-like behavior that contains a single physical engagement of the robot agent $\mathcal{A}_k$ with the object $\mathcal{O}_k^i$. The subtask $\mathcal{S}_k^i$ is composed of the following three basic manipulation phases:

1) *Approach:* The agent $\mathcal{A}_k$ moves toward the object $\mathcal{O}_k^i$.
2) *Manipulate:* The agent physically interacts with the object (e.g., by grasping, pushing, holding, transporting, etc.).
3) *Leave:* The agent $\mathcal{A}_k$ disengages with the object $\mathcal{O}_k^i$ and returns to its home position.

IEEE Proof

Fig. 2. Robot-centric task progress monitor and three-phase structure of subtasks.

For our developments, we consider a class of sequential manipulation tasks that can be represented as the concatenation of multiple subtasks. This task is conducted by single/multiple robotic systems in AM, and a human may enter the scene to (possibly) contribute/modify the task's progress. To command this robotic system to reactively (i.e., with awareness of the human's actions) perform the manipulation task, there are three problems that must be solved.

1) *Model:* The total sequential manipulation task should be decomposed into multiple robot-centric action sequences, each of which is constructed with several subtasks.
2) *Context:* The system should be able to compute the changes introduced to the task by the human.
3) *Control:* The robotic agents should reactively adapt to the new context and complete the manipulation task.

In the following sections, we present our proposed methodology to address these problems, whose overall control architecture is conceptually depicted in Fig. 1.

III. Task Progress Monitor of Sequential Manipulation Tasks

A. Three-Phase Structure of the Substask

The sequential manipulation tasks considered in this study consist of a series of basic actions, each serving distinct purposes. Some actions directly influence the configuration of objects, while others facilitate transitions between these physical interactions. The proposed *three-phase structure* is constructed with *approach*, *manipulate*, and *leave* to model this sequential procedure (as shown in Fig. 2). These three phases are denoted for the kth robot during the ith subtask as $\mathcal{S}_k^{i1}$, $\mathcal{S}_k^{i2}$, and $\mathcal{S}_k^{i3}$, respectively. The corresponding features are $d_k^{i1} = \|\mathbf{x}_k - \mathbf{o}_k^i\|$, $d_k^{i2} = \|\mathbf{o}_k^i - \mathbf{o}_k^{i*}\|$, and $d_k^{i3} = \|\mathbf{x}_k - \mathbf{y}_k^i\|$. Define the feature domain D_k for the action sequence of the kth robot and D for the whole system as follows:

$$D_k = \left\{ d_k^{ij} \,|\, i=0,1,\ldots,N_k, j=1,2,3 \right\},\ D = \bigcup_{k=1}^{n-1} D_k. \quad (1)$$

We quantify these phases with the Boolean vector $\mathbf{r}_k^i = [r_k^{i1}, r_k^{i2}, r_k^{i3}]^\top \in \mathbb{R}^3$, which is computed based on geometric feedback as follows:

$$\mathbf{r}_k^i = \left[\mathbf{1}(d_k^{i1}), \quad \mathbf{1}(d_k^{i2}), \quad \mathbf{1}(d_k^{i3})\right]^\top \quad (2)$$

where $\mathbf{1}(x)$ is the Heaviside step function defined as

$$\mathbf{1}(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

The Boolean coordinates of $\mathbf{r}_k^i$ activate under the following condition.

1) When the robot pose $\mathbf{x}_k$ is different from the object's pose $\mathbf{o}_k^i$ (this provides a signal that disables once the robot reaches with the object, and that is used for the *approach* phase).
2) When the object pose $\mathbf{o}_k^i$ is different from the target object pose $\mathbf{o}_k^{i*}$ (this provides a signal that disables once the object reaches the desired configuration, and that is used for the *manipulate* phase).
3) When the robot pose $\mathbf{x}_k$ is different from the home pose $\mathbf{y}_k^i$ (this provides a signal that disables once the robot reaches its respective home position, and that is used for the *leave* phase).

To properly synchronize the phases among different robots collaborating on the task, we introduce the vector $\bar{\mathbf{r}}_k^i = [\bar{r}_k^{i1}, \bar{r}_k^{i2}, \bar{r}_k^{i3}]^\top$, and which is computed with a functional Boolean operation g as follows:

$$\bar{r}_k^{ij} = g(\mathbf{\Phi}_k^i) : \mathbb{R}^{3(n-2)} \mapsto \mathbb{R}^3, \mathbb{R}^{L_{kij}} \mapsto \mathbb{R}^1 \quad (4)$$

$$\mathbf{\Phi}_k^i = \left[\mathbf{1}(\phi_1), \ldots, \mathbf{1}(\phi_l), \ldots, \mathbf{1}(\phi_{L_{kij}})\right]^\top, \phi_l \in D/D_k \quad (5)$$

where L_{kij} is the number of the features relative to $\mathcal{S}_k^{ij}$. The vector $\bar{\mathbf{r}}_k^i$ is related to the geometric features of *approach*, *manipulate*, and *leave* of the other $n-1$ collaborating agents including human. Its purpose is to signal the kth robot $\mathcal{A}_k$ to complete its current phase based on task assignment or collaboration conditions. Especially, $\mathbf{r}_k^0 = \mathbf{0}_3$ as well as $\bar{\mathbf{r}}_k^0$ indicates the condition to start the first subtask in the collaboration. For example, consider the subtask of $\mathcal{A}_1$ that requires to simultaneously move an object along $\mathcal{A}_2$. When the robot $\mathcal{A}_1$ has already grasped the object, its *approach* phase should not end although the respective feature $r_1^{i1} = 0$. To complete this phase, the robot $\mathcal{A}_1$ needs to wait for $\mathcal{A}_2$ to grasp the other end, i.e., $\bar{r}_1^{i1} = \mathbf{1}(\|\mathbf{x}_2 - \mathbf{o}_2^q\|) \to 0$, therefore, the *manipulate* phase can begin, and both robots jointly move the object. Section V provides multiple examples of the construction of the two Boolean vectors $\mathbf{r}_k^i$ and $\bar{\mathbf{r}}_k^i$.

B. Task Progress Monitor

Based on the proposed three-phase model, locate the task progress in the schedule at the phase with the highest priority. For that, we introduce the *phase priority vector* for each subtask $\mathcal{S}_k^i$, for $i = 0, 1, \ldots, N_k$ and $k = 1, \ldots, n-1$ as follows:

$$\mathbf{p}_k^i = a_k^i \cdot \mathbf{\Omega}_k^i(\mathbf{r}_k^i + \bar{\mathbf{r}}_k^i) = [p_k^{i1}, p_k^{i2}, p_k^{i3}]^\top \in \mathbb{R}^3 \quad (6)$$

where

$$\mathbf{\Omega}_k^i = \begin{bmatrix} \neg r_k^{(i-1)3} \wedge \neg \bar{r}_k^{(i-1)3} & 0 & 0 \\ 0 & \neg r_k^{i1} \wedge \neg \bar{r}_k^{i1} & 0 \\ 0 & 0 & \neg r_k^{i2} \wedge \neg \bar{r}_k^{i2} \end{bmatrix}. \quad (7)$$

Specifically, we set $\mathbf{r}_k^0 = \mathbf{0}_3$. The diagonal matrix $\mathbf{\Omega}_k^i$ guarantees that the previous phase is always completed before a new phase is activated. The Boolean coordinates of $\mathbf{p}_k^i$ signify the active phase with the highest priority, denoted by assigning a value of 1. It is designed such that only one coordinate can have a value of 1 at any given time. This priority vector needs to be computed for all robots and subtasks. Upon evaluating the values of these coordinates, robot $\mathcal{A}_k$ then proceeds to execute the phase with the highest priority.

To schedule all subtasks in the correct order, we introduce the *active coefficient*, which is computed with the following unified rules:

$$a_k^i = \begin{cases} \neg r_k^{(i-1)2} \wedge r_k^{i2}, & \mathbf{o}_k^{i*} \text{ is the latest target of } \mathcal{O}_k^i \\ 0, & \mathcal{O}_k^i \text{ has a new target.} \end{cases} \quad (8)$$

IEEE Proof

The proposed active coefficient a_k^i guarantees that a subtask can only be activated if its goal has not been reached and its previous subtask has been completed. The subtask would be silenced from the whole sequence if the goal of the object related to it has been updated in the future.

Then, our task progress monitor in the equation below integrates all phase priorities into a long column vector. The only coordinate with a value of 1 is the specific phase that must be conducted in the action sequence of the robot.

$$\mathbf{m}_k = \left[\mathbf{p}_k^{1\top}, \ldots, \mathbf{p}_k^{i\top}, \ldots, \mathbf{p}_k^{N_k\top}\right]^\top \in \mathbb{R}^{3N_k}. \tag{9}$$

This method allows the system to reschedule or extend the task by moving or adding phase priority vectors in the monitor.

IV. Human-Aware Reactive Task Planning

A. Mode Switching From HM to AM

To switch mode at a proper timing, it is imperative to define the mode trigger, denoted as h. It can be designed to suit the specific needs of the system. In this article, we focus on two key factors: safety and effectiveness, which are encapsulated in the design of the mode trigger as $h = h_s + h_e$.

For safety considerations, the speed of $\mathcal{A}_n$ is taken into account, resulting in $h_s = \mathbf{1}(\|\dot{\mathbf{x}}_n\|)$. This formulation ensures that mode switching to HM is activated ($h = 1$) only when a human agent is detected in motion. In terms of effectiveness, we evaluate the contribution of $\mathcal{A}_n$ to the manipulation goal. This is determined by examining the time derivative of features related to the "manipulate" phase. Specifically, we define that $\mathcal{A}_n$ contributes to the subtask when $\dot{d}_k^{i2} < 0$ or any $\dot{\phi}_l < 0$, $(l = 1, \ldots, L_{ki2})$. In essence, if $\mathcal{A}_n$'s actions lead to a decrease in d_k^{i2} or any ϕ_l, the mode trigger is activated to switch to HM. Otherwise, HM remains inactive ($h = 0$), indicating that AM continues. To sum up, the mode trigger is

$$h = h_s + h_e = \mathbf{1}(\|\dot{\mathbf{x}}_n\|) + \bigvee_{l=1}^{L_{ki2}} \mathbf{1}(-\dot{\phi}_l) + \mathbf{1}(-\dot{d}_k^{i2}). \tag{10}$$

Note that, if a phase depends on the NOT operation of a feature ($\neg\mathbf{1}(\phi_l)$), which means the opposite change of the feature (the distance gets larger) is required, the logic is $\dot{\phi}_l > 0 \rightarrow -\dot{\phi}_l < 0 \rightarrow \mathbf{1}(-\dot{\phi}_l) = 0 \rightarrow \neg\mathbf{1}(-\dot{\phi}_l) = 1$.

Next, the process of the mode switching from HM to AM is stated. The influence caused by $\mathcal{A}_n$ can be modeled by introducing a human coefficient agent matrix $\mathbf{\Omega}_H$ to extend (6) and (9) to the following forms:

$$\hat{\mathbf{p}}_k^i = a_k^i \mathbf{\Omega}_H \mathbf{\Omega}_k^i (\mathbf{r}_k^i + \bar{\mathbf{r}}_k^i) = [\hat{p}_k^{i1}, \hat{p}_k^{i2}, \hat{p}_k^{i3}]^\top \in \mathbb{R}^3 \tag{11}$$

$$\hat{\mathbf{m}}_k = \left[\hat{\mathbf{p}}_k^{1\top}, \ldots, \hat{\mathbf{p}}_k^{i\top}, \ldots, \hat{\mathbf{p}}_k^{N_k\top}\right]^\top \in \mathbb{R}^{3N_k}. \tag{12}$$

Given the phase priority computed by (11), during the mode switching, $\mathcal{A}_k$ should conduct the phase $\mathcal{S}_k^{ij_k^*}$ to end the interrupted ith subtask, then conduct $\mathcal{S}_k^{i_k^*}$ that is newly activated after $\mathcal{S}_k^i$ has been completed, where

$$j_k^* = \underset{j=1,2,3}{\operatorname{argmax}}\, \hat{p}_k^{ij}, \text{ and } i_k^* = \underset{i}{\operatorname{argmax}}\, a_k^i. \tag{13}$$

All robotic agents execute their cut-points in chronological order in the task progress monitor.

TABLE I
Truth Table of Three Human Agent Coefficient Matrices for Reactive Tasks

(a) Ω_H^1

f_1, f_2	j_k^*
1, 1	1
1, 0	2
0, -	3

(b) Ω_H^2

f_1, f_2, f_3	j_k^*
1, 1, -	1
1, 0, 1	2
1, 0, 0	3
0, -, -	3

(c) Ω_H^3

f_1, f_2, f_3, f_4	j_k^*
1, 1, -, 0	1
1, 0, 1, 0	2
1, 0, 0, 0	3
0, -, -, 0	3
-, 0, -, 1	1
-, 1, -, 1	3

B. Reactive Task Design

Given that the environment has been altered by the human agent during HM, the robotic system must update its understanding of the environment and task progress to identify appropriate cut-points for switching back to AM. Consequently, it is crucial to select suitable *environmental features* that accurately describe the environment and aid in designing the necessary functions. Denote the distance between the human agent and $\mathcal{O}_k^i$ as $d_n^{ki} = \|\mathbf{x}_n - \mathbf{o}_k^i\|$. In our study, we select four key features, that is, $f_1 = \mathbf{1}(d_k^{i2})$, $f_2 = \mathbf{1}(d_k^{i1})$, $f_3 = \mathbf{1}(d_n^{ki})$, and $f_4 = \mathbf{1}(i - i_k^*)$. The truth values of these features indicate the following: $\mathcal{S}_k^i$ has not been completed, the kth robot does not engage with $\mathcal{O}_k^i$, the human agent does not engage with $\mathcal{O}_k^i$, and a previous subtask is active again. In many cases, these features are essential for updating the system's understanding of the environment and determining the appropriate transition points between AM and HM.

Given specific combinations of these selected environmental features, various human agent coefficient matrices (introduced in (11) to describe the human influence) can be designed to achieve different desired reactive tasks. The complexity of the reactive tasks dictates the number of environmental features to be utilized. Each combination maps to a single phase, serving as a cut-point in the interrupted subtask. Boolean operations can then be applied to derive a specific solution for $\mathbf{\Omega}_H$. In our research, we present three examples, with their corresponding truth tables displayed in Table I.

1) Feature Vector is $\mathbf{f} = [f_1, f_2]^\top$: When the interrupted $\mathcal{S}_k^i$ is incomplete ($f_1 = 1$), $\mathcal{A}_k$ should *approach* $\mathcal{O}_k^i$ if $\mathcal{A}_k$ is not engaged with $\mathcal{O}_k^i$ ($f_2 = 1$), and *manipulate* $\mathcal{O}_k^i$ if $\mathcal{A}_k$ is engaged with $\mathcal{O}_k^i$ ($f_2 = 0$). When the interrupted $\mathcal{S}_k^i$ is complete ($f_1 = 0$), $\mathcal{A}_k$ should *leave*. The solution is given by

$$\mathbf{\Omega}_H^1 = \begin{bmatrix} f_1 f_2 & 0 & 0 \\ 0 & f_1(1 - f_2) & 0 \\ 0 & 0 & 1 - f_1 \end{bmatrix}. \tag{14}$$

2) Feature Vector is $\mathbf{f} = [f_1, f_2, f_3]^\top$: Besides the aforementioned functions, the privilege of $\mathcal{A}_n$ is taken into account. If both $\mathcal{A}_n$ and $\mathcal{A}_k$ are touching $\mathcal{O}_k^i$ simultaneously ($f_2 = 0, f_3 = 0$), $\mathcal{A}_k$ should *leave*. If only $\mathcal{A}_n$ is touching $\mathcal{O}_k^i$ ($f_3 = 0$), $\mathcal{A}_k$ should *approach*. The solution is given by

$$\mathbf{\Omega}_H^2 = \begin{bmatrix} f_1 f_2 f_3 & 0 & 0 \\ 0 & f_1(1 - f_2) f_3 & 0 \\ 0 & 0 & 1 - f_1 + f_1(1 - f_3) \end{bmatrix}. \tag{15}$$

3) Feature Vector is $\mathbf{f} = [f_1, f_2, f_3, f_4]^\top$: In sequential manipulation tasks, the success of a subtask often depends on previous

IEEE Proof

Fig. 3. Task progress monitor and the corresponding environmental features under AM of single-arm stacking task.

subtasks. If a previous subtask is reactivated ($f_4 = 1$), $\mathcal{A}_k$ should redo the subtask after withdrawing the current one. Otherwise ($f_4 = 0$), $\mathcal{A}_k$ follows the scheduled subtasks. Furthermore, when $f_4 = 1$, if $\mathcal{A}_k$ is engaged with $\mathcal{O}_k^i$ ($f_2 = 0$), $\mathcal{A}_k$ should *approach* the pose where $\mathcal{A}_k$ gets engaged with $\mathcal{O}_k^i$ to safely release $\mathcal{O}_k^i$ before execute the active subtask, and if $\mathcal{A}_k$ is available ($f_2 = 1$), it should *leave*. The solution is given by

$$\mathbf{\Omega}_H^3 = \begin{bmatrix} \omega_1 & 0 & 0 \\ 0 & \omega_2 & 0 \\ 0 & 0 & \omega_3 \end{bmatrix}, \text{where}$$

$$\begin{aligned} \omega_1 &= f_1 f_2 f_3 (1 - f_4) + (1 - f_2) f_4 \\ \omega_2 &= f_1 (1 - f_2) f_3 (1 - f_4) \\ \omega_3 &= (1 - f_1)(1 - f_4) + f_1 (1 - f_3)(1 - f_4) + f_2 f_4. \end{aligned} \quad (16)$$

V. Cases of Study

It is assumed that the manipulation can be automatically conducted by robot, which is reasonable in a smart factory adopting HRI techniques for intelligent and resilient systems. To validate our proposed method, we conducted studies on three manipulation tasks: single-arm stacking, dual-arm assembly, and dual-arm packing. We used MediaPipe (open-source at[1]) to detect human gestures in the environment. For more details, refer to the video[2] and the attached document.

A. AM of the Tasks

The task progress monitor and the corresponding environmental features of three sequential manipulations are illustrated in Figs. 3–5. In these figures, $d(a, b)$ represents the Euclidean distance between a and b. Three distances correspond to r_k^{i1}, r_k^{i2}, and r_k^{i3}. In the Gantt-like chart, (ST)i-j in the row for $\mathcal{A}_k$ represents S_k^{ij}, which is the jth phase of the ith subtask for $\mathcal{A}_k$.

1) Single-Arm Stacking: In this case, there are two agents ($n = 2$): the robot $\mathcal{A}_1$ and the human agent $\mathcal{A}_2$. Since the task is executed by a single arm, the collaboration is modeled as $\bar{\mathbf{r}}_1^i = \mathbf{0}_3$. This task involves placing blue blocks in a straight line at the bottom and positioning yellow blocks at the top between adjacent blue blocks, comprising a series of pick-and-place subtasks. The AM for single-arm stacking facilitates the addition of new objects in the environment by continuously detecting if any object is not at its target pose.

2) Dual-Arm Assembly: In this case, there are three agents ($n = 3$): the left robotic arm $\mathcal{A}_1$, and right robotic arm $\mathcal{A}_2$, and the human agent $\mathcal{A}_3$. This task aims to arrange yellow blocks

[1][Onlilne]. Available: https://github.com/google/mediapipe
[2][Onlilne]. Available: https://vimeo.com/886803313?share=copy

Fig. 4. Task progress monitor and the corresponding environmental features under AM of dual-arm assembly task. The top row is $\mathcal{A}_1$ and the bottom row is $\mathcal{A}_2$.

in a specific order, ensuring the purple patterns form a heart. Following this, a green bar is threaded through the middle of the yellow blocks, which are then pushed tightly together. Finally, two robotic arms move the green bar holding the yellow blocks to a frame. This task comprises heterogeneous subtasks, including pick-and-place and pushing actions.

The manipulation patterns of $\mathcal{A}_1$ are pick-and-place, pick-and-place, hold, push, and pick-and-place; the manipulation patterns of $\mathcal{A}_2$ are pick-and-place, push, and pick-and-place. Define the function for contact detection at $\mathbf{o}_k^i$ as follows:

$$c(\mathbf{o}_k^i) = \bigvee_{q=1, q \neq k}^{n} \mathbf{1}(\|\mathbf{x}_q - \mathbf{o}_k^i\|). \quad (17)$$

$c(\mathbf{o}_k^i) = 1$ means there is at least one agent contacts $\mathbf{o}_k^i$. The collaboration is modeled as follows:

$$\begin{aligned} &\bar{\mathbf{r}}_1^0 = \mathbf{0}_3, \quad \bar{\mathbf{r}}_1^1 = \mathbf{0}_3 \\ &\bar{\mathbf{r}}_1^2 = \begin{bmatrix} 0 & 0 & \mathbf{1}(\|\mathbf{o}_2^1 - \mathbf{o}_2^{1*}\|) \wedge \mathbf{1}(\|\mathbf{x}_2 - \mathbf{y}_2^1\|) \end{bmatrix}^\top \\ &\bar{\mathbf{r}}_1^3 = \begin{bmatrix} 0 & c(\mathbf{o}_2^3) & 0 \end{bmatrix}^\top, \quad \bar{\mathbf{r}}_1^4 = \mathbf{0}_3, \quad \bar{\mathbf{r}}_1^5 = \begin{bmatrix} c(\mathbf{o}_2^3) & 0 & 0 \end{bmatrix}^\top \\ &\bar{\mathbf{r}}_2^0 = \begin{bmatrix} 0 & 0 & \mathbf{1}(\|\mathbf{o}_1^2 - \mathbf{o}_1^{2*}\|) \wedge \mathbf{1}(\|\mathbf{x}_1 - \mathbf{y}_1^2\|) \end{bmatrix}^\top \\ &\bar{\mathbf{r}}_2^1 = \begin{bmatrix} 0 & 0 & \mathbf{1}(\|\mathbf{x}_1 - \mathbf{o}_1^3\|) \wedge \mathbf{1}(\|\mathbf{o}_1^3 - \mathbf{o}_1^{3*}\|) \end{bmatrix}^\top \\ &\bar{\mathbf{r}}_2^2 = \mathbf{0}_3, \quad \bar{\mathbf{r}}_2^3 = \begin{bmatrix} c(\mathbf{o}_1^5) & 0 & 0 \end{bmatrix}^\top. \end{aligned}$$

3) Dual-Arm Packing: In this case, there are three agents ($n = 3$): the left robotic arm $\mathcal{A}_1$, and right robotic arm $\mathcal{A}_2$, and the human agent $\mathcal{A}_3$. This task involves placing a long linear soft object into a compact box through sequential subtasks. Due to the object's elasticity, it tends to pop up from the box after being placed. In addition, when the robot finishes placing the object and opens the gripper to release it, the gripper is pressed by the elastic object and the walls of the box, causing the object to be pulled out. To address this challenge, the robots collaborate by alternately performing pick-and-place and fixing actions to successfully complete the task.

The manipulation patterns of $\mathcal{A}_1$ are pick-and-place, pick-and-place, fix, and fix; the manipulation patterns of $\mathcal{A}_2$ are fix, fix, pick-and-place, and fix. The collaboration is modeled as follows (refer to the supplementary document for more details):

IEEE Proof

Fig. 5. Task progress monitor and the corresponding environmental features under AM of dual-arm packing task. The top row is $\mathcal{A}_1$ and the bottom row is $\mathcal{A}_2$.

Fig. 6. Stacking Case (f). (1) Robot movement: Moving $\mathcal{O}_1^4$ to the target. (2)–(4) Human movement: Take over $\mathcal{O}_1^4$ and put it at the target. (5) Robot reaction: Leave $\mathcal{O}_1^4$ and reset.

$$\bar{\mathbf{r}}_1^0 = \mathbf{0}_3, \quad \bar{\mathbf{r}}_1^1 = \begin{bmatrix} 0 & c(\mathbf{o}_2^1) & 0 \end{bmatrix}^\top$$
$$\bar{\mathbf{r}}_1^2 = \begin{bmatrix} 0 & \neg\mathbf{1}(\|\mathbf{x}_2 - \mathbf{o}_2^1\|) \wedge c(\mathbf{o}_2^2) & 0 \end{bmatrix}^\top$$
$$\bar{\mathbf{r}}_1^3 = \begin{bmatrix} 0 & \neg\mathbf{1}(\|\mathbf{x}_2 - \mathbf{o}_2^2\|) \wedge c(\mathbf{o}_2^3) \wedge \mathbf{1}(\|\mathbf{o}_2^3 - \mathbf{o}_2^{3*}\|) & 0 \end{bmatrix}^\top$$
$$\bar{\mathbf{r}}_1^4 = \begin{bmatrix} 0 & \neg\mathbf{1}(\|\mathbf{x}_2 - \mathbf{o}_2^3\|) \wedge c(\mathbf{o}_2^4) \wedge \mathbf{1}(\|\mathbf{o}_2^4 - \mathbf{o}_2^{4*}\|) & 0 \end{bmatrix}^\top$$
$$\bar{\mathbf{r}}_2^0 = \begin{bmatrix} 0 & 0 & \mathbf{1}(\|\mathbf{o}_1^1 - \mathbf{o}_1^{1*}\|) \wedge c(\mathbf{o}_1^1) \end{bmatrix}^\top$$
$$\bar{\mathbf{r}}_2^1 = \begin{bmatrix} 0 & \neg\mathbf{1}(\|\mathbf{x}_1 - \mathbf{o}_1^1\|) \wedge c(\mathbf{o}_1^2) \wedge \mathbf{1}(\|\mathbf{o}_1^2 - \mathbf{o}_1^{2*}\|) & 0 \end{bmatrix}^\top$$
$$\bar{\mathbf{r}}_2^2 = \begin{bmatrix} 0 & \neg\mathbf{1}(\|\mathbf{x}_1 - \mathbf{o}_1^2\|) \wedge c(\mathbf{o}_1^3) \wedge \mathbf{1}(\|\mathbf{o}_1^3 - \mathbf{o}_1^{3*}\|) & 0 \end{bmatrix}^\top$$
$$\bar{\mathbf{r}}_2^3 = \begin{bmatrix} 0 & \neg\mathbf{1}(\|\mathbf{x}_1 - \mathbf{o}_1^3\|) \wedge c(\mathbf{o}_1^4) & 0 \end{bmatrix}^\top$$
$$\bar{\mathbf{r}}_2^4 = \begin{bmatrix} 0 & \mathbf{1}(\|\mathbf{x}_1 - \mathbf{y}_1^4\|) & 0 \end{bmatrix}^\top$$

where two typical collaborative behaviors are as follows.

1) $\bar{\mathbf{r}}_1^2$: $\mathcal{A}_1$ ends the *manipulate* phase of $\mathcal{S}_1^2$ and then starts the *leave* phase of $\mathcal{S}_1^2$ if $\mathcal{A}_2$ safely leaves $\mathcal{O}_2^1$ and there is an agent touching $\mathcal{O}_2^2$.
2) $\bar{\mathbf{r}}_1^3$: $\mathcal{A}_1$ ends the *manipulate* phase of $\mathcal{S}_1^3$ and starts the *leave* phase of $\mathcal{S}_1^3$ if $\mathcal{A}_2$ safely leaves $\mathcal{O}_2^2$, there is an agent touching $\mathcal{O}_2^3$, and $\mathcal{O}_2^3$ reaches its target position.

B. Experimental Results of Reactive Tasks

We conducted a series of interactions with the robotic system, designed for different purposes, to validate the effectiveness of our proposed method. The robots' reactions were guided by the human agent coefficient matrices, which were designed based on the feature vector $\mathbf{f} = [f_1, f_2, f_3, f_4]^\top$. The task progress monitors and three features for each of the three cases are shown in Figs. 7–9. In these figures, $\lambda_1 = \|\mathbf{o}_k^i - \mathbf{o}_k^{i*}\|$, $\lambda_2 = \|\mathbf{x}_k^i - \mathbf{o}_k^i\|$, and $\lambda_3 = \|\mathbf{x}_n^i - \mathbf{o}_k^i\|$. The black dashed lines represent the 0–1

Fig. 7. Task progress monitor and features of single-arm stacking under HM. object-i is manipulated in $\mathcal{S}_1^i$.

boundaries. The detailed interactions are as follows, where the lowercase letter in the round brackets corresponds the sub-figure in Fig. 7–9.

1) Single-Arm Stacking: This includes the following cases. *S-(a)*: $\mathcal{A}_2$ approaches to pick $\mathcal{O}_1^1$, while $\mathcal{A}_1$ is also approaching $\mathcal{O}_1^1$. Then, $\mathcal{A}_2$ places $\mathcal{O}_1^1$ at its target location. The reaction is that $\mathcal{A}_1$ *leaves* $\mathcal{O}_1^1$ and proceeds to execute $\mathcal{S}_1^2$. *S-(b)*: $\mathcal{A}_2$ adds $\mathcal{O}_1^4$ when $\mathcal{A}_1$ is holding $\mathcal{O}_1^2$. The reaction is that $\mathcal{A}_1$ resumes to *manipulate* $\mathcal{O}_1^2$. *S-(c)*: $\mathcal{A}_2$ moves $\mathcal{O}_1^2$ away from its target place when $\mathcal{A}_1$ is moving $\mathcal{O}_1^3$. The reaction is that $\mathcal{A}_1$ *approaches* the initial grasped pose to release $\mathcal{O}_1^3$, and then, redoes $\mathcal{S}_1^2$. *S-(d)*: $\mathcal{A}_2$ adds

IEEE Proof

Fig. 8. Task progress monitors and features of dual-arm assembly under HM. object-k-i is manipulated in $\mathcal{S}_k^i$.

Fig. 9. Task progress monitors and features of dual-arm packing under HM. object-k-i is manipulated in $\mathcal{S}_k^i$.

$\mathcal{O}_1^5$ when $\mathcal{A}_1$ is placing $\mathcal{O}_1^2$ at its target. The reaction is that $\mathcal{A}_1$ *leaves* $\mathcal{O}_1^2$. *S-(e)-1*: $\mathcal{A}_2$ hands over $\mathcal{O}_1^3$ to $\mathcal{A}_1$. The reaction is that $\mathcal{A}_1$ *approaches* to grasp $\mathcal{O}_1^3$. *S-(e)-2*: $\mathcal{A}_2$ leaves after $\mathcal{A}_1$ grasps $\mathcal{O}_1^3$. The reaction is that $\mathcal{A}_1$ resume *manipulating* $\mathcal{O}_1^3$. *S-(f)-1*: $\mathcal{A}_2$ takes over $\mathcal{O}_1^4$ from $\mathcal{A}_1$. The reaction is that $\mathcal{A}_1$ *leaves* $\mathcal{O}_1^4$ by opening its grippers. *S-(f)-2*: $\mathcal{A}_2$ immediately places $\mathcal{O}_1^4$ at its target. The reaction is that $\mathcal{A}_1$ resume *leaving*. Two interactions of S-(f) are demonstrated in the Fig. 6. *S-(g)*: $\mathcal{A}_2$ holds $\mathcal{O}_1^5$ to fix it, while $\mathcal{A}_1$ places $\mathcal{O}_1^5$ at its target. The reaction is that $\mathcal{A}_1$ resume *leaving*.

2) Dual-Arm Assembly: This includes the following cases. *A-(a)*: $\mathcal{A}_3$ approaches to touch $\mathcal{O}_1^3$ with the intention of taking over from $\mathcal{A}_1$. The reaction is that $\mathcal{A}_1$ *leaves* $\mathcal{O}_1^3$ by opening the grippers and $\mathcal{A}_2$ remains stationary. *A-(b)*: $\mathcal{A}_3$ replaces $\mathcal{A}_1$ to push $\mathcal{O}_1^4$ to its target position, while $\mathcal{A}_1$ is holding $\mathcal{O}_1^3$. The reaction is that $\mathcal{A}_1$ skips $\mathcal{S}_1^4$ and proceeds directly to $\mathcal{S}_1^5$, and $\mathcal{A}_2$ remains stationary. *A-(c)*: $\mathcal{A}_3$ first exits the environment, and then, moves away $\mathcal{O}_2^2$ from its target. The reaction is that $\mathcal{A}_1$ *approaches* to grasp $\mathcal{O}_1^3$ and $\mathcal{A}_2$ *leaves* $\mathcal{O}_2^3$ to redo pushing $\mathcal{O}_2^2$. Since the cut-point of $\mathcal{A}_1$ occurs earlier than $\mathcal{A}_2$, $\mathcal{A}_2$ waits in $\mathcal{S}_2^{32}$ for $\mathcal{A}_1$ to finish $\mathcal{S}_1^{31}$ (grasping $\mathcal{O}_1^3$). After $\mathcal{A}_1$ completes $\mathcal{S}_1^{31}$, $\mathcal{A}_2$ proceeds to $\mathcal{S}_2^{31}$ to place $\mathcal{O}_2^3$ back in its initial position. Since the initial pose of $\mathcal{O}_2^3$ is the same as the grasped pose, $\mathcal{A}_2$ skips the redundant motions and directly performs $\mathcal{S}_2^{33}$, then redoes $\mathcal{S}_2^2$ to push $\mathcal{O}_2^2$ to the target. This case demonstrates our task progress monitor can intelligently provide an appropriate solution (phase) for execution from a long-term perspective based on the local solution provided by the human agent coefficient matrix.

3) Dual-Arm Packing: This includes the following cases. *P-(a)-1*: $\mathcal{A}_3$ passes $\mathcal{O}_1^2$ to $\mathcal{A}_1$. The reaction is that $\mathcal{A}_1$ *approaches* to grasp $\mathcal{O}_1^2$ and $\mathcal{A}_2$ remains stationary. *P-(a)-2*: After $\mathcal{A}_1$ grasps $\mathcal{O}_1^2$, $\mathcal{A}_3$ exits the environment. The reaction is that $\mathcal{A}_1$ resumes placing (*manipulating*) $\mathcal{O}_1^2$ at the target and $\mathcal{A}_2$ continues fixing $\mathcal{O}_2^1$. *P-(b)*: $\mathcal{A}_3$ touches the $\mathcal{O}_1^2$ while $\mathcal{A}_1$ is moving it to the target. The reaction is that $\mathcal{A}_1$ *leaves* by opening the grippers and $\mathcal{A}_2$ remains stationary. *P-(c)-1*: $\mathcal{A}_3$ replaces $\mathcal{A}_2$ to fix $\mathcal{O}_2^1$. The reaction is that $\mathcal{A}_1$ *leaves* to perform $\mathcal{S}_1^2$ and $\mathcal{A}_2$ waits for $\mathcal{A}_1$ before *leaving* to reset. For $\mathcal{A}_2$, there are the following four stages [highlighted in shadows in Fig. 9(c)] for analysis.

a) Black box: the human is moving, $h_s = 1$.
b) Red box: $\dot{d}_1^{13} > 0 \rightarrow \mathbf{1}(\dot{d}_1^{13}) \rightarrow h_e = 1$.
c) Green box: $\dot{d}_1^{21} < 0 \rightarrow \mathbf{1}(-\dot{d}_1^{21}) \rightarrow h_e = 1$.
d) Yellow box: $\dot{d}_1^{22} < 0 \rightarrow \mathbf{1}(-\dot{d}_1^{22}) \rightarrow h_e = 1$.

The HM ends when $\mathcal{A}_1$ places $\mathcal{O}_1^2$ at its target, resulting in $h = 0$. This case shows the flexibility of our mode trigger in guiding multiple robots, balancing safety and task effectiveness. *P-(c)-2*: $\mathcal{A}_3$ exits the environment, while $\mathcal{A}_1$ is holding $\mathcal{O}_1^2$ at its target and $\mathcal{A}_2$ is fixing $\mathcal{O}_2^2$. The reaction is that $\mathcal{A}_1$ *leaves* $\mathcal{O}_1^2$ to proceed to the next subtask and $\mathcal{A}_2$ continues fixing (*manipulating*) $\mathcal{O}_2^2$ before leaving.

C. Summary of Human Intentions

These HRI cases illustrate the essential functions required to appropriately respond to human agent intentions in real-world applications.

1) In cases *S-(e)-1* and *P-(a)-1*, the human agent hands over objects to facilitate the robot's grasping at a better pose or with an enhanced view.
2) The human agent manipulates objects in advance to save time or enhance collaboration in cases *S-(a)*, *A-(b)*, and *P-(c)-1-*$\mathcal{A}_1$.
3) The human agent takes over grasped objects in cases *S-(f)-1*, *A-(a)*, and *P-(b)*.
4) In cases *S-(e)-2*, *P-(a)-2*, *P-(c)-1-*$\mathcal{A}_2$, and *P-(c)-2*, the human agent passes by or exits the environment.
5) The human agent signals the system to redo a subtask to correct results or improve performance in cases *S-(c)* and *A-(c)*.
6) The human agent aids by holding and fixing objects in cases *S-(f)-2* and *S-(g)*.
7) The human agent introduces new objects into the environment in cases *S-(b)* and *S-(d)*.

IEEE Proof

TABLE II
COMPARISON OF THREE HUMAN AGENT COEFFICIENT MATRICES—SINGLE-ARM STACKING

Case	Feature	$\mathrm{diag}(\Omega_H^1)/j_k^*$	$\mathrm{diag}(\Omega_H^2)/j_k^*$	$\mathrm{diag}(\Omega_H^3)/j_k^*$
S-(a)	0 1 1 0	[0 0 1]/3	[0 0 1]/3	[0 0 1]/3
S-(b)	1 0 1 0	[0 1 0]/2	[0 1 0]/2	[0 1 0]/2
S-(c)	1 0 1 1	[0 1 0]/2	[0 1 0]/2	[1 0 0]/1
S-(d)	0 0 1 0	[0 0 1]/3	[0 0 1]/3	[0 0 1]/3
S-(e)-1	1 1 0 0	[1 0 0]/1	[1 0 0]/1	[1 0 0]/1
S-(e)-2	1 0 1 0	[0 1 0]/2	[0 1 0]/2	[0 1 0]/2
S-(f)-1	1 0 0 0	[0 1 0]/2	[0 0 1]/3	[0 0 1]/3
S-(f)-2	0 1 0 0	[0 0 1]/3	[0 0 1]/3	[0 0 1]/3
S-(g)	0 0 0 0	[0 0 1]/3	[0 0 1]/3	[0 0 1]/3

TABLE III
COMPARISON OF THREE HUMAN AGENT COEFFICIENT MATRICES—DUAL-ARM ASSEMBLY

Case	Feature	$\mathrm{diag}(\Omega_H^1)/j_k^*$	$\mathrm{diag}(\Omega_H^2)/j_k^*$	$\mathrm{diag}(\Omega_H^3)/j_k^*$
A-(a)-$\mathcal{A}_1$	1 0 0 0	[0 1 0]/2	[0 0 1]/3	[0 0 1]/3
A-(b)-$\mathcal{A}_1$	0 0 1 0	[0 0 1]/3	[0 0 1]/3	[0 0 1]/3
A-(c)-$\mathcal{A}_1$	1 1 1 0	[1 0 0]/1	[1 0 0]/1	[1 0 0]/1
A-(c)-$\mathcal{A}_2$	1 0 1 1	[0 1 0]/2	[0 1 0]/2	[0 0 1]/1

TABLE IV
COMPARISON OF THREE HUMAN AGENT COEFFICIENT MATRICES—DUAL-ARM PACKING

Case	Feature	$\mathrm{diag}(\Omega_H^1)/j_k^*$	$\mathrm{diag}(\Omega_H^2)/j_k^*$	$\mathrm{diag}(\Omega_H^3)/j_k^*$
P-(a)-1-$\mathcal{A}_1$	1 1 0 0	[1 0 0]/1	[1 0 0]/1	[1 0 0]/1
P-(a)-2-$\mathcal{A}_1$	1 0 1 0	[0 1 0]/2	[0 1 0]/2	[0 1 0]/2
P-(b)-$\mathcal{A}_1$	1 0 0 0	[0 1 0]/2	[0 0 1]/3	[0 0 1]/3
P-(c)-1-$\mathcal{A}_1$	0 1 1 0	[0 0 1]/3	[0 0 1]/3	[0 0 1]/3
P-(c)-1-$\mathcal{A}_2$	0 1 0 0	[0 0 1]/3	[0 0 1]/3	[0 0 1]/3
P-(c)-2-$\mathcal{A}_1$	0 0 1 0	[0 1 0]/3	[0 0 1]/3	[0 0 1]/3
P-(c)-2-$\mathcal{A}_2$	1 0 1 0	[0 1 0]/2	[0 1 0]/2	[0 1 0]/2

These cases demonstrate the diverse and critical roles a human agent plays in enhancing robotic performance and ensuring seamless collaboration in various manipulation tasks.

D. Comparison of Human Agent Coefficient Matrix

The environmental features and the results of three human agent coefficient matrices [see (14)–(16)] are summarized in Tables II–IV. Occasionally, these three matrices yield identical results. However, in complex reactive tasks, such as reactivating a past subtask (S-(c), A-(c)-$\mathcal{A}_2$) and when the human agent intends to take over (S-(f)-1, A-(a)-$\mathcal{A}_1$, P-(b)-$\mathcal{A}_1$), additional environmental features should be incorporated. This ensures that robots respond with greater intelligence and security, thereby improving the overall task success rate. In some cases, the system must determine the priority between the human agent (who always has the highest priority in the studied cases) and robots when both are interacting with an object, particularly considering human strength (e.g., robots should not release the object if it is too heavy). To address this, the weight and volume of objects to be manipulated should be factored into the priority determination.

E. Independence From Specific Actions

Due to the unpredictable nature of human agent movements, understanding human intentions or translating them into robotic actions is challenging. This unpredictability can cause disruptions in task progress tracking under HM. A significant advantage of the proposed mode switching strategy is its independence from specific tasks and actions. To highlight this advantage, in this section, all robots operate under HM, meaning AM is active whenever a robot is moving. Consequently, reactive tasks are executed in AM. The features used to construct the task progress monitor, mode trigger, and reactive tasks are based on temporal and spatial relationships (e.g., distances between agents and objects, distances between objects and their targets, and completion status of a subtask) rather than specific actions (e.g., pick-and-place, push, and fix). Based on the aforementioned discussion, our solution ensures that unrecorded actions do not lead to unexpected consequences, as validated by the experiments shown in Fig. 10.

Fig. 10. (1) Planned actions in AM (shown in Fig. 5): The left robot $\mathcal{A}_1$ grasps and places the object $\mathcal{O}_1^1$ in the box, the right robot $\mathcal{A}_2$ fixes the object $\mathcal{O}_2^1$, $\mathcal{A}_1$ leaves $\mathcal{O}_1^1$ without damage; Human intervention in HM: The human agent stays in the environment and keeps fixing after grasping and placing the object $\mathcal{O}_2^1$ in the box. (2) $\mathcal{A}_1$ grasps $\mathcal{O}_1^2$ and places it in the box. (3) $\mathcal{A}_2$ fixes $\mathcal{O}_2^2$ to help $\mathcal{A}_1$ leave $\mathcal{O}_1^2$ without damage.

The independence from specific tasks limits our strategy in terms of applications, as its effectiveness heavily relies on the system's capabilities under AM and the principles defined by users. For instance, if a human agent introduces a new object into the environment, the mode switching strategy will fail if the system cannot recognize the new object under AM. In such cases, the mode switching strategy becomes irrelevant. Therefore, it is essential for the human agent to either upgrade the AM capabilities or avoid expecting the robot to handle new objects to prevent this type of failure.

F. Ablation Experiments

In the existing works, when it comes to the execution of AM/HM mode switching (especially from HM to AM), the interaction is as simple as pick-and-place so that the robotic systems do not need intelligent reactions. In the more complicated tasks, the robotic system and the human operator can inflexibly wait for each other to finish the current action. Meanwhile, most research efforts typically focus on rescheduling entire tasks in response to critical failures. However, this approach is often unsuitable when only local adjustments are needed. Moreover, the solutions of switching paths or halting the robot to avoid collisions with human agents often neglect the implicit collaborative intentions of human agents. Furthermore, most works limits the interaction between the human operator with single arm. Therefore, the comprehensive and necessary HM/AM switching execution and the interaction between human operator with single/multiple robots remains an open problem. Our work addresses this gap by exploring more flexible and efficient execution strategies. We achieve this by identifying optimal timing cut-points through a proposed task progress monitoring monitor.

To sum up, there are the following three critical functionalities of our algorithm:

1) accurate task progress tracking considering the collaboration among robots and the randomness of the human agent's movements;
2) reactive task planner for flexible disengagement of the intervened subtask and a suitable timing cut-point on the task progress monitor to realize specific customer-designed effects;

IEEE Proof

TABLE V
COMPARISON WITH OTHER WORKS

	Agent number		Intervened subtask		Mode trigger	
	Single	Multiple	Current	Past	Safe	Efficient
[29]	✓	✗	✓	✗	✓	✗
[30]	✓	✗	✓	✗	✓	✗
[31]	✓	✓	✓	✗	✓	✗
[32]	✓	✓	✓	✓	✓	✗
Ours	✓	✓	✓	✓	✓	✓

Fig. 11. (1) Human agent $\mathcal{A}_3$ disengages from replacing the left robot $\mathcal{A}_1$ and initiates interaction with the right robot $\mathcal{A}_2$ by reactivating its previous subtask. (2) With our method, $\mathcal{A}_1$ regrasps the green bar in preparation for $\mathcal{A}_2$'s redoing of the object pushing task. (3) Without accounting for the collaboration of robots, the system resumes operation at an inopportune moment, resulting in task failure.

Fig. 12. (1) Robot $\mathcal{A}_1$ is going to place $\mathcal{O}_1^3$ onto $\mathcal{O}_1^1$ and $\mathcal{O}_1^2$, while the human agent $\mathcal{A}_2$ moves $\mathcal{O}_1^2$ away. (2) Without our method, $\mathcal{A}_1$ continues the intervened subtask leading to the task failure. (3) With our method, $\mathcal{A}_1$ can properly disengage the intervened subtask to redo the reactivated subtask.

Fig. 13. (1) Human agent $\mathcal{A}_3$ replaces the left robot $\mathcal{A}_1$ to put the object in the box. (2) Without our method, a collision between $\mathcal{A}_1$ and $\mathcal{A}_3$ occurs. (3) With our method, $\mathcal{A}_3$ can safely replace $\mathcal{A}_1$ to work in the environment and the right robot $\mathcal{A}_2$ can resume operation.

3) intelligent mode trigger balancing safety and efficiency to facilitate seamless switching between AM and HM considering human movements and intentions.

This section first makes a comparison with other works about these three critical functionalities (as listed in Table V), and then, presents a series of ablation experiments aimed at validating the efficacy and significance of our algorithm to facilitate seamless switching between AM and HM.

The results of these experiments, illustrated in Figs. 11–14, serve to demonstrate the robustness and versatility of our proposed approach.

Fig. 14. (1) The left robot $\mathcal{A}_1$ grasps the object and places it in the box while the human agent $\mathcal{A}_3$ replaces the right robot $\mathcal{A}_2$ to fix the object. (2) Without our method, the robots pause for safety whenever $\mathcal{A}_3$ is detected leading to a stuck manipulation. (3) With our method, the HM can be activated when a static human agent contributes to the task.

1) Accurate Task Progress Tracking Considering the Collaboration Among Robots: As shown in Fig. 11, the human agent $\mathcal{A}_3$ initially replaces the left robot $\mathcal{A}_1$ to grasp the green bar, subsequently releasing it to displace an object, thereby triggering the reactivation of a prior subtask for the right robot $\mathcal{A}_2$. In the absence of our method: the green bar lacks a fixed endpoint following $\mathcal{A}_2$'s gripper release, attributed to $\mathcal{A}_1$'s inability to recover from intervention; during the reactivated subtask execution by $\mathcal{A}_2$ (pushing the right yellow block toward the center), an unexpected bias occurs in the position of the green bar, indicated by deviations from vertical midlines; ultimately, $\mathcal{A}_2$ fails to grasp the green bar. Leveraging our method for accurate task progress tracking considering the influence among the robots, $\mathcal{A}_1$ adeptly regrasps the green bar by strategically timing its actions based on the task progress monitor, although at this stage, $\mathcal{A}_3$ does not interact with it. This allows $\mathcal{A}_2$ to relinquish control of the green bar and resume pushing the yellow block, ensuring smooth task continuity.

2) Reactive Task Planner for Flexible Disengagement of the Intervened Subtask: As shown in Fig. 12, the robot $\mathcal{A}_1$ is tasked with placing $\mathcal{O}_1^3$ onto both $\mathcal{O}_1^1$ and $\mathcal{O}_1^2$, while human agent $\mathcal{A}_2$ moves $\mathcal{O}_1^2$ (already positioned at its target) aside. Leveraging our method to design reactive tasks based on as many relevant environmental features as possible, $\mathcal{A}_1$ can return the grasped object to its initial position, a crucial step for successfully completing the interrupted task. In the absence of our method, the results obtained from the human agent coefficient matrices, defined by two or three environmental features in *S-(c)*, reveal that insufficient environmental features guide $\mathcal{A}_1$ to resume manipulation within the interrupted subtask. This invariably leads to task failure, with $\mathcal{O}_1^3$ toppling from the surface of $\mathcal{O}_1^1$.

3) Intelligent Mode Trigger Balancing Safety and Efficiency: Human motion detection is fundamental for ensuring safety in HRI tasks. As illustrated in Fig. 13, our mode trigger can activate HM and pause the robots to prevent collisions when a human agent is detected moving.

However, it is inefficient for the robots to remain stationary as long as a human agent is detected, as a static human agent can contribute to manipulation or collaboration. As shown in Fig. 14, the left robot $\mathcal{A}_1$ grasps the object and places it in the box while the human agent $\mathcal{A}_3$ replaces the right robot $\mathcal{A}_2$ to fix the object. In the absence of our method, the robots pause for safety whenever $\mathcal{A}_3$ is detected, leading to a halt in the manipulation process. In addition, it is impractical for all robots to switch modes simultaneously. Leveraging our method to assess a static

IEEE Proof

human agent's contribution to manipulation or collaboration, each robot can switch modes at different times, resulting in an intelligent mode-switching process. In Fig. 14(a) and (b), our mode trigger guides $\mathcal{A}_1$ to resume its subtasks AM since the static $\mathcal{A}_3$ does not contribute to its task, while $\mathcal{A}_2$ remains in HM because $\mathcal{A}_3$ is assisting $\mathcal{A}_2$ by replacing it to fix the object. In Fig. 14(c), our mode trigger guides $\mathcal{A}_2$ switches from HM to AM because $\mathcal{A}_3$ does not contribute to $\mathcal{A}_2$'s current subtask.

VI. Conclusion

In this study, we introduced a novel approach for transitioning robotic systems between short-term human-aware mode (HM) and long-horizon automatic mode (AM) while executing customer-designed reactive tasks. We developed a task progress monitor for sequential manipulation, allowing continuous tracking despite human interventions, by decomposing tasks into robot-centric action sequences and subtasks. Our three-phase frameworkcomprising approach, manipulate, and leave phases-models these subtasks, considering all robotic agents involved. We designed a mode trigger algorithm to balance safety and efficiency, activating HM in various scenarios, including static human collaboration. A human agent coefficient matrix facilitates reactive task execution during mode transitions. Extensive experimentation validated our method's efficacy across various tasks and robotic systems, accommodating diverse human intentions. Our approach excels in identifying suitable cut-points within the AM timeline with minimal computational overhead. However, task completion post-reaction relies on AM functionality, suggesting future research could explore automated task segmentation using multimodal data and large language models for enhanced task understanding.

References

[1] S. Li et al., "Proactive human–robot collaboration: Mutual-cognitive, predictable, and self-organising perspectives," *Robot. Comput.-Integr. Manuf.*, vol. 81, 2023, Art. no. 102510.

[2] J. Lv et al., "A multimodality scene graph generation approach for robust human–robot collaborative assembly visual relationship representation," *IEEE Trans. Ind. Inform.*, vol. 20, no. 3, pp. 3242–3251, Mar. 2024.

[3] P. Paral, A. Chatterjee, A. Rakshit, and S. K. Pal, "Extended target tracking in human–robot coexisting environments via multisensor information fusion: A heteroscedastic Gaussian process regression based approach," *IEEE Trans. Ind. Inform.*, vol. 19, no. 9, pp. 9877–9886, Sep. 2023.

[4] L. Wang, S. Liu, C. Cooper, X. V. Wang, and R. X. Gao, "Function block-based human-robot collaborative assembly driven by brainwaves," *CIRP Ann.*, vol. 70, no. 1, pp. 5–8, 2021.

[5] N. Jarrasse, V. Sanguineti, and E. Burdet, "Slaves no longer: Review on role assignment for human–robot joint motor action," *Adapt. Behav.*, vol. 22, no. 1, pp. 70–82, 2014.

[6] C. Wang and J. Zhao, "Role dynamic assignment of human-robot collaboration based on target prediction and fuzzy inference," *IEEE Trans. Ind. Inform.*, vol. 20, no. 1, pp. 471–481, Jan. 2024.

[7] L. Johannsmeier and S. Haddadin, "A hierarchical human-robot interaction-planning framework for task allocation in collaborative industrial assembly processes," *IEEE Robot. Automat. Lett.*, vol. 2, no. 1, pp. 41–48, Jan. 2017.

[8] A. Roncone, O. Mangin, and B. Scassellati, "Transparent role assignment and task allocation in human robot collaboration," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 1014–1021.

[9] A. Nurbayeva, A. Shintemirov, and M. Rubagotti, "Deep imitation learning of nonlinear model predictive control laws for safe physical human-robot interaction," *IEEE Trans. Ind. Inform.*, vol. 19, no. 7, pp. 8384–8395, Jul. 2023.

[10] A. Capitanelli, M. Maratea, F. Mastrogiovanni, and M. Vallati, "On the manipulation of articulated objects in human–robot cooperation scenarios," *Robot. Auton. Syst.*, vol. 109, pp. 139–155, 2018.

[11] E. Matheson, R. Minto, E. G. Zampieri, M. Faccio, and G. Rosati, "Human–robot collaboration in manufacturing applications: A review," *Robotics*, vol. 8, no. 4, 2019, Art. no. 100.

[12] BSI Group, *Robots and Robotic Devices–Collaborative Robots*, BSI Standards Publication: London, UK, 2016.

[13] D. Andronas et al., "Towards seamless collaboration of humans and high-payload robots: An automotive case study," *Robot. Comput.-Integr. Manuf.*, vol. 83, 2023, Art. no. 102544.

[14] R. Jahanmahin, S. Masoud, J. Rickli, and A. Djuric, "Human-robot interactions in manufacturing: A survey of human behavior modeling," *Robot. Comput.-Integr. Manuf.*, vol. 78, 2022, Art. no. 102404.

[15] M. K. M. Rabby, A. Karimoddini, M. A. Khan, and S. Jiang, "A learning-based adjustable autonomy framework for human–robot collaboration," *IEEE Trans. Ind. Inform.*, vol. 18, no. 9, pp. 6171–6180, Sep. 2022.

[16] T. Kaupp and A. Makarenko, "Measuring human-robot team effectiveness to determine an appropriate autonomy level," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2008, pp. 2146–2151.

[17] T. Stouraitis, I. Chatzinikolaidis, M. Gienger, and S. Vijayakumar, "Online hybrid motion planning for dyadic collaborative manipulation via bilevel optimization," *IEEE Trans. Robot.*, vol. 36, no. 5, pp. 1452–1471, Oct. 2020.

[18] Á. Martínez-Tenor and J.-A. Fernández-Madrigal, "Smoothly adjustable autonomy for the low-level remote control of mobile robots that is independent of the navigation algorithm," in *Proc. Mediterranean Conf. Control Automat.*, 2015, pp. 1071–1078.

[19] L. Han, W. Xu, P. Kang, and H. Yuan, "Unified neural adaptive control for multiple human–robot–environment interactions," *IEEE Trans. Ind. Inform.*, vol. 17, no. 2, pp. 1166–1175, Feb. 2021.

[20] A. M. Zanchettin, A. Casalino, L. Piroddi, and P. Rocco, "Prediction of human activity patterns for human–robot collaborative assembly tasks," *IEEE Trans. Ind. Inform.*, vol. 15, no. 7, pp. 3934–3942, Jul. 2019.

[21] W. Li, D. Sadigh, S. S. Sastry, and S. A. Seshia, "Synthesis for human-in-the-loop control systems," in *Proc. Int. Conf. Tools Algorithms Construction Anal. Syst.*, 2014, pp. 470–484.

[22] F. Fusaro, E. Lamon, E. De Momi, and A. Ajoudani, "A human-aware method to plan complex cooperative and autonomous tasks using behavior trees," in *Proc. IEEE Int. Conf. Humanoids*, 2021, pp. 522–529.

[23] M. Braun, Y. Cheng, and M. Tomizuka, "Human-aware robot task planning with robot execution time estimation," *IFAC-PapersOnLine*, vol. 55, no. 41, pp. 181–186, 2022.

[24] D. Rodriguez-Guerra, G. Sorrosal, I. Cabanes, and C. Calleja, "Human-robot interaction review: Challenges and solutions for modern industrial environments," *IEEE Access*, vol. 9, pp. 108557–108578, 2021.

[25] G. Notomista, S. Mayya, S. Hutchinson, and M. Egerstedt, "An optimal task allocation strategy for heterogeneous multi-robot systems," in *Proc. 18th Eur. Control Conf.*, 2019, pp. 2071–2076.

[26] T. Yu, J. Huang, and Q. Chang, "Optimizing task scheduling in human-robot collaboration with deep multi-agent reinforcement learning," *J. Manuf. Syst.*, vol. 60, pp. 487–499, 2021.

[27] F. Flacco, "The tasks priority matrix: A new tool for hierarchical redundancy resolution," in *Proc. IEEE-RAS 16th Int. Conf. Humanoid Robots*, 2016, pp. 1–7.

[28] S. Hong, G. Park, and Y. Lee, "Robust design and task-priority control for rescue robot HURCULES," *J. Field Robot.*, vol. 40, no. 5, pp. 1151–1171, 2023.

[29] S. Li, D. Park, Y. Sung, J. A. Shah, and N. Roy, "Reactive task and motion planning under temporal logic specifications," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 12618–12624.

[30] S. Hangl, E. Ugur, S. Szedmák, J. H. Piater, and A. Ude, "Reactive, task-specific object manipulation by metric reinforcement learning," in *Proc. Int. Conf. Adv. Robot.*, 2015, pp. 557–564.

[31] N. M. Ceriani, A. M. Zanchettin, P. Rocco, A. Stolt, and A. Robertsson, "Reactive task adaptation based on hierarchical constraints classification for safe industrial robots," *IEEE/ASME Trans. Mechatron.*, vol. 20, no. 6, pp. 2935–2949, Dec. 2015.

[32] Z. Zhou, D. J. Lee, Y. Yoshinaga, S. B. Balakirsky, D. Guo, and Y. Zhao, "Reactive task allocation and planning for quadrupedal and wheeled robot teaming," in *Proc. IEEE 18th Int. Conf. Automat. Sci. Eng.*, 2021, pp. 2110–2117.

IEEE Proof