IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Towards Efficient Human-Robot Collaboration with Robust Plan Recognition and Trajectory Prediction

Yujiao Cheng[1], Liting Sun[1], Changliu Liu[2], and Masayoshi Tomizuka[1]

Abstract— **Human-robot collaboration (HRC) is becoming increasingly important as the paradigm of manufacturing is shifting from mass production to mass customization. The introduction of HRC can significantly improve the flexibility and intelligence of automation. To efficiently finish tasks in HRC systems, the robots need to not only predict the future movements of human, but also more high-level plans, i.e., the sequence of actions to finish the tasks. However, due to the stochastic and time-varying nature of human collaborators, it is quite challenging for the robot to efficiently and accurately identify such task plans and respond in a safe manner. To address this challenge, we propose an integrated human-robot collaboration framework. Both plan recognition and trajectory prediction modules are included for the generation of safe and efficient robotic motions. Such a framework enables the robots to perceive, predict and adapt their actions to the human's work plan and intelligently avoid collisions with the human. Moreover, by explicitly leveraging the hierarchical relationship between plans and trajectories, more robust plan recognition performance can be achieved. Physical experiments were conducted on an industrial robot to verify the proposed framework. The results show that the proposed framework could accurately recognize the human workers' plans and thus significantly improve the time efficiency of the HRC team even in the presence of motion classification noises.**

I. Introduction

As the emphasis of manufacturing is shifting from mass production to mass customization, the demands for flexible automation keep increasing. Human-robot collaboration (HRC), as an effective and efficient way to enhance the flexibility, has attracted lots of attention both in industry and academia in the past decade. The idea of HRC is to let robots work safely and collaboratively with humans in a shared space. To achieve this, robots should be equipped with various capabilities from fundamental skills, such as perception of human activities, to higher-level social skills, including reasoning about intentions and collaboration [1].

Collaboration between humans and intelligent robots can be categorized into three levels: 1) low-level collision avoidance, 2) middle-level efficient cooperation with task plan recognition and trajectory prediction, and 3) high-level collaboration mode selection and automatic task assignments. Many researches have been conducted for all these three levels. The first category regards human as moving obstacles and designs algorithms to let the robot avoid collisions with human. The third category, e.g., [2], studies the task assignment algorithms in peer-to-peer human-robot interaction where humans and robots work as partners.

In this paper, we focus on the second category which includes three key elements. The first one is human trajectory prediction which aims to predict the continuous human movement used for safe robot trajectory planning. It is different from discrete intention recognition such as [3]–[6]. Many approaches have been proposed for continuous trajectory prediction, from early attempts such as Kalman filter and particle filter [7] [8] to recent effort such as recurrent neural networks [9], inverse reinforcement learning [10] [11] and semi-adaptable neural network [12]. A human-aware robotic system which incorporated both motion predictions and trajectory planning has also been presented in [13].

The second element is human plan recognition. It aims to recognize what plan the human is doing given the observed trajectories and his/her influence to the objects. Human activity/plan recognition has attracted a great amount of effort. Some work focused on deep learning frameworks with RGBD images as inputs [14] [15] [16]. Typically, the features selected mainly focus on human, for instance, the body pose, hand positions, motion information and histogram of oriented gradients (HOG). No information about the objects of interaction are included. However, the objects can provide rich information for inferring what the human is doing via the intrinsic hierarchy among actions, motions and the objects. Hence, in this paper, we explore such hierarchy to design more robust plan recognition algorithm.

With good plan recognition module and trajectory prediction module, the third key element for efficient human-robot collaboration is the behavior/action generation for robots, i.e., the planner. Although there are many robot behavior generation approaches [6] [1], [17]–[20], most of them focus on the action level rather than the plan level.

We propose an integrated HRC framework which includes both trajectory prediction and plan recognition. At trajectory planning level, robots take future human trajectory into account to avoid potential collisions, which improves safety. At task planning level, robots perceive the human's actions, infer the human plan and adapt to the human's actions in advance to boost the collaboration efficiency. The proposed framework advantages HRC in three aspects. First, the robot is more responsive to the human's plans, particularly when there might be change of plans in the human's operations. By using our proposed plan recognition method, the robot can quickly recognize the human's new plan, and adapt its actions accordingly. Experimental results showed that the average task completion time is significantly reduced, i.e., more efficient HRC can be achieved. Second, our system is robust with respect to noises in the model inputs and errors in the intermediate steps such as motion classification. We combine a long short-term memory (LSTM) network with algorithms based on Bayesian inference instead of end-to-end learning. Moreover, a set of hierarchical relationships among trajectories, motions, actions, plans and tasks are explicitly defined and utilized in the plan recognition algorithm. This not only helps improve the robustness of the algorithm, but also reduce the dimension of the problem, and enhance its generalization ability using less data.

The key contributions of this work are:

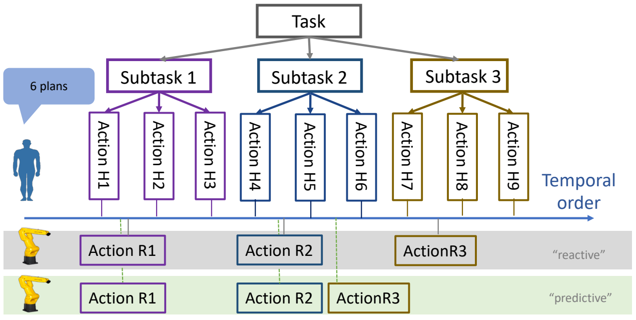
1) We propose a robot system interleaving prediction/recognition and adaptation at both trajectory and task planning level.
2) We propose a robust plan recognition algorithm by leveraging the hierarchical relationship between the plan and the trajectories.
3) We provide physical experiments to evaluate the whole system in a desktop assembly task using an industrial robot arm. Results demonstrate that the system improves the efficiency of HRC, and the plan recognition algorithm is robust in the presence of noises such as false motion classification.

II. An Integrated Plan Recognition and Trajectory Prediction Framework

In this work, we focus on enabling better HRC systems via plan recognition and trajectory prediction. The terminologies we used in this paper are defined as follows:

- **Trajectory**: a time series of the joint positions of an agent (either a human or a robot) in Cartesian space. It represents the continuous movements of an agent.
- **Motion**: A discrete variable/label to represent different types/patterns of trajectories. For instance, typical motions in factory scenarios include “Fetching”, “Picking”, “Screwing” and “Taping”. Different trajectories can be generated to perform the same motion pattern.
- **Action**: A paired discrete variable/label including a motion pattern and the target object to act on, i.e., action=$\{\text{motion}, \text{object}\}$. For example, we can define “action 1=$\{\text{Fetching}, \text{a screwdriver}\}$”, “action 2=$\{\text{Taping}, \text{a bunch of cables}\}$” and so on.
- **Subtask**: A subtask is an element of completing a larger *task* (defined below), whose initial states and the goal states do not depend on other subtasks. It might be implemented with several sequences of actions, depending on their orders.
- **Plan**: A plan is comprised of a sequence of ordered subtasks. It represents the preferences to finish a *task* (defined below). Different orders of actions in different plans come from either orders of subtasks, or the orders of actions within subtasks.
- **Task**: A task represents the work to be conducted by agents. It specifies the initial states, the goal states and the participants. A task can be decomposed into a set of subtasks and executed via a variety of plans.

We use Fig. 1 and a **desktop assembly** example to illustrate the hierarchical relationship of the terminologies. A task (“desktop assembly”) can be decomposed into three subtasks (“installing a CPU fan”, “installing a system fan”, “taping cables”). Suppose each of them has a unique action order, then the permutation of three subtasks totally generates six different plans, all of which are stored in the plan library as action[1] sequences of the human and the robot. Furthermore, within each action, the motion can be executed by infinite many (theoretically) trajectories. Trajectory prediction is to forecast the future movement of the human, thus the robot can make safe trajectory planning avoiding potential collisions. Plan recognition is to choose the correct plan in human’s mind, which is to choose the predefined action sequence in the plan library. As shown in Fig. 1, without plan recognition, the robot (the “reactive” robot) can only acquire its next action after the human finishes some key actions (such as Action H1, Action H4 and Action H7), while with a plan recognition, the robot (the “predictive” robot) can foresee the future actions of the human and execute its following actions in advance to boost the efficiency of the collaboration.



Plan 1: Subtask 1-> Subtask 2-> Subtask 3; Plan 2: Subtask 1-> Subtask 3-> Subtask 2
Plan 3: Subtask 2-> Subtask 1-> Subtask 3; Plan 4: Subtask 2-> Subtask 3-> Subtask 1
Plan 5: Subtask 3-> Subtask 1-> Subtask 2; Plan 6: Subtask 3-> Subtask 2-> Subtask 1

Fig. 1: A hierarchical and temporal decomposition of a task.

However, human plans are not directly observable for the robot. The only observable variables of humans are their trajectories, which means that the robot has to infer and reason about the probable plans by observing the trajectories of humans. Such diversity, un-observability, and time-varying characteristic of the human plans create great challenges for the HRC systems. It requires the robot to 1) quickly and reliably recognize the plans of humans, and 2) responsively adapt its own behavior in a safe and predictive manner to ensure efficient and seamless collaboration.

To address these two challenges, we propose an integrated HRC framework, the architecture of which is shown in Fig. 2. It includes both offline database and online modules. Online modules include a perception module (sensors and perception algorithms), a plan recognition module, a trajectory prediction module, a planner, a motion control module and

[1] In the desktop assembly example, Action H1-H9 are “fetching the cpu fan”, “receiving the screwdriver A”, “screwing the cpu fan”, “fetching the system fan”, “receiving the screwdriver B”, “screwing the system fan”, “taping the cables”, “receiving scissors”, and “cutting the tape”. Action R1-R3 are “delivering screwdriver A”, “delivering screwdriver B” and “delivering scissors”

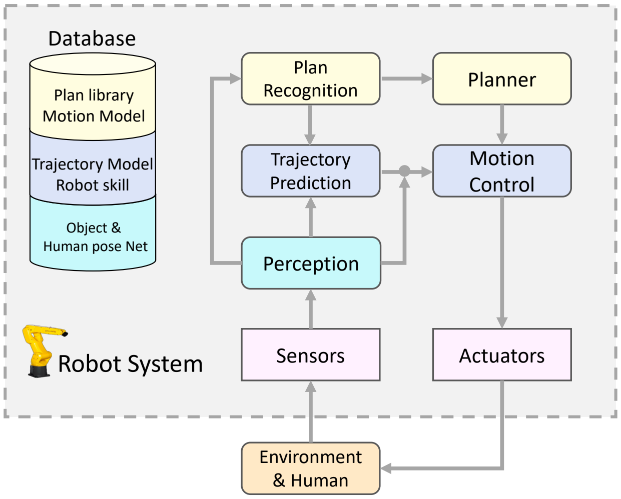


Fig. 2: The architecture of the proposed integrated HRC framework

the actuators (the robot). *Perception Module* takes visual information as inputs and outputs the 3D positions of objects as well as the 3D human poses. *Plan Recognition Module* is a key module in our proposed framework. It aims to identify the action being executed by the human and infer human's plan by observing the trajectories of their key joints. The action estimate will be sent to both the planner module and the trajectory prediction module. *Planner Module* assigns the next action (a motion-object pair) to the robot based on the current states, the recognized plan and the current action of the human. The action command from the planner will be sent to the motion control module. *Trajectory Prediction Module* aims to predict the future trajectories of the human. Instead of directly predicting the future trajectories based on only current and historical human trajectories, we leverage the action labels from the plan recognition module. *Motion Control Module* includes two controllers: an efficiency controller and a safety controller, as in [21]. The efficiency controller is a long-term global controller to assure the efficiency of robot, and the safety controller is a short-term local controller for real-time safety under uncertainties.

III. The Plan Recognition Algorithm

As discussed in Section II, to enable efficient and seamless human-robot collaboration, the robot needs to quickly and reliably recognize the plan executed by a human, and safely adapt its behavior. However, the diversity of human plans for the same task, the time-inconsistent or time-varying characteristic of humans, as well as the un-observability of plans have posed great challenges for accurate and timely plan recognition. To address these challenges, we propose a plan recognition algorithm based on both deep learning techniques and Bayesian inference as shown in Fig. 3. Moreover, we explicitly take advantage of the hierarchical relationships among "trajectory", "action", "subtask" and "plan", and design the plan recognition into three mutually compensated steps for better plan recognition. The four steps of plan recognition are: motion classification, target object estimation, plan inference and posterior action correction.

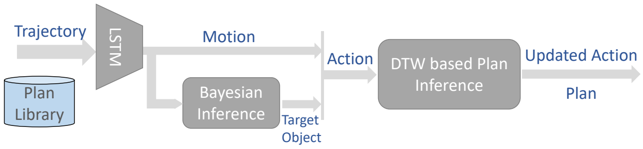


Fig. 3: The architecture of the plan recognition module.

The proposed plan recognition algorithm has advantages in two aspects. First, the dimension of action space is reduced via the hierarchical combination of motion and object. Notice that an action is defined as a pair of motion and object, and the set of candidate objects for different motions can be quite different. Hence, a hierarchical combination of motion classification and target object estimation can help significantly reduce the dimension of the classification problem compared to direct action classification. Second, more robust recognition performance can be achieved via the posterior update step of the action based on the plan information. With this step, prior domain knowledge in regard to the relationships of plan and actions is fully utilized to help reduce the sensitivity of learning based methods to noises.

A. *Motion Classification*

Motion classification aims to categorize different motions given segments of trajectories of the human's key joints. Long-short-term-memory (LSTM) neural networks have been extensively proved to be an effective approach to model the dynamics and dependencies in sequential data. Hence, we design a LSTM recurrent neural network for motion classification. The input data is the human pose from the *Perception* module. More specifically, in an assembly task, the input vector at time step k is $\mathbf{x}_k{=}\{\mathbf{w}_k, \mathbf{h}_k\}$, where $\mathbf{w}_k$ is the wrist position in the world frame and $\mathbf{h}_k$ are the velocities of selected key points on the human fingers. The output at time step k is a motion label $m_k{\in}\{1, 2, ...n_m\}$, where $n_m{\in}\mathbb{N}$ is the number of motions. The LSTM is trained using the "Motion Model" database in Fig. 2.

B. *Target Object Estimation*

Given the classified motion labels and a history of human pose, Bayesian inference is commonly used to update the beliefs on different target objects, e.g. [22].

Let o_k be an object at time step k, $\mathcal{O}$ be the object set, $m_{1:k}$ be the historical motion labels, and $h_{1:k}$ be the historical human poses. Then we need to obtain the robot's beliefs on the object, i.e., a probability $P(o_k|h_{1:k}, m_{1:k})$. Applying the Markov assumption, the following equation holds:

$$
\begin{aligned}
&P(o_k|h_{1:k}, m_{1:k}) \propto \\
&P(m_k|o_k, h_{k-1}, m_{k-1}) \cdot P(h_k|o_k, h_{k-1}, m_k) \\
&\sum_{o_{k-1}\in\mathcal{O}} P(o_k|h_{k-1}, m_{k-1}, o_{k-1}) \cdot P(o_{k-1}|h_{1:k-1}, m_{1:k-1})
\end{aligned}
$$

We compute the $P(h_k|o_k, h_{k-1}, m_k)$ with an assumption that humans are optimizing some value function as [23] suggests. Then a Boltzmann policy can be applied:

$$P(h_k|o_k, h_{k-1}, m_k) \propto \exp(\beta V_g(h_k; o_k))$$

where V_g is the value function. We model V_g for each motion as a function of distance and velocity.

To compute $P(m_k|o_k, h_{k-1}, m_{k-1})$ and $P(o_k|h_{k-1}, m_{k-1}, o_{k-1})$, we impose conditional independence assumption of m_k and h_{k-1} given o_k and m_{k-1}, and conditional independence assumption of o_k and h_{k-1} given o_{k-1} and m_{k-1}. Then, with predefined or learned models of $P(m_k|m_{k-1}, o_k)$ and $P(o_k|m_{k-1}, o_{k-1})$, $P(o_k|h_{1:k}, m_{1:k})$ can be updated iteratively.

C. Plan Inference

With results from motion classification and object estimation, we can uniquely determine a sequence of actions by observing the human trajectories. Note that a plan is a sequence of subtasks, and each subtask is represented by one action or an ordered sequence of actions. Hence, a plan can be uniquely represented by a temporal sequence of actions. Therefore, we first build a plan library offline in the *Database* where each plan is represented by a reference sequence of actions. Then we utilize the reference sequences to online infer potential plans based on Bayes' rule,

$$P(g|a_{1:k}) \propto P(a_{1:k}|g)P(g),$$

where $P(g)$ is a prior belief of plan g, and $P(g|a_{1:k})$ is a posterior belief based on the likelihood of observed action sequence $a_{1:k}$ given plan g. Similarly, with Boltzmann policy, the likelihood of the action trajectory can be defined as

$$P(a_{1:k}|g) \propto \exp(-d(a_{1:k}; g)),$$

where the function d is a distance function measuring the similarity between observed action sequence (A, namely $a_{1:k}$) and the reference action sequence (R) of the plan g. The larger the distance is, the less likely the human is following the plan [24]. We adopt the open-end dynamic time warping (OE-DTW) algorithm [25] to calculate d. This algorithm is to best match the query sequence to a reference sequence and calculate the dissimilarity between the matched portion. Given a reference time series $R = (r_1, r_2, ..., r_N)$ and a query sequence $A = (a_1, a_2, ..., a_M)$, the OE-DTW distance between A and R is calculated via minimizing the dynamic time warping distances (DTW) between A and any references R^j truncated from reference R at point $j = 1 : N$.

$$D_{OE}(A, R) = \min_{j=1,...,N} D_{DTW}(A, R^j).$$

Here is a short introduction to DTW. The indices of the two series will be mapped through ϕ_t and ψ_t, $t = 1, 2, ..., T$, that satisfy the following constraints [25]:

- Boundary condition: $\phi_1 = 1$, $\psi_1 = 1$ and $\phi_T = N$, $\psi_T = M$
- Monotonic conditions: $\phi_{t-1} \leq \phi_t$ and $\psi_{t-1} \leq \psi_t$
- Continuity conditions: $\phi_t - \phi_{t-1} \leq 1$, and $\psi_t - \psi_{t-1} \leq 1$
- Local slope constraints: certain step patterns are allowed.

The optimal $\hat{\Phi} = (\hat{\phi}_t, \hat{\psi}_t)$ minimizes the distance between the two warped time series:

$$(\hat{\phi}_t, \hat{\psi}_t) = arg \min_{\phi_t, \psi_t} \sum_{t=1}^{T} \frac{d(r_{\phi_t}, a_{\psi_t}) m_{t,\Phi}}{\sum_t m_{t,\Phi}},$$

where $d(\cdot, \cdot)$ is any distance function and $m_{t,\Phi}$ is a local weighting coefficient. Therefore, the dynamic time warping distance between A and R is

$$D_{DTW}(A, R) = \sum_{t=1}^{T} \frac{d(r_{\hat{\phi}_t}, a_{\hat{\psi}_t}) m_{t,\Phi}}{\sum_t m_{t,\Phi}}.$$

D. Posterior Action Correction

As we obtain the posterior estimate of the plan g^*, the best matched reference sequence R^{*j^*} is also obtained. We correct the action label estimate a_k by retrieving the action in the best matched reference plan as follows,

$$a_k^{post} = R^*(j^*) = r^*_{j^*}.$$

This step is of key importance to reduce the sensitivity of the learning models to noises, so that the robustness of the plan recognition can be improved. The effectiveness of this step is verified in experiments.

IV. Human Trajectory Prediction

To avoid collisions between a human and a robot, the future human trajectory is required to be considered in the safety controller to generate safety constraints. We leverage two inputs: 1) the human pose estimates from *perception module* and 2) the action labels from the *plan recognition module*. The human transition model is approximated by a feedforward neural network, the output layer of which is adapted online using recursive least square parameter adaptation algorithm (RLS-PAA) to address the challenges regarding the time-varying characteristic of human trajectories. For more details of this approach, one can refer to our previous work [12].

V. The Planner

In this section, we will present the *Planner* module and the workflow of the proposed framework. As shown in Fig. 2, the output of the *Plan recognition* module is sent to the *Planner* to generate commands for the *Motion control* module. With the identified plan and the human action estimate, the *Planner* module acquires the next action of the robot from the plan library, and sets the goal states of the *Motion control* module to generate safe and executable trajectories.

Since the plan recognition results are probabilistic, we need to design a decision-making mechanism to decide on the robot actions. There are two cases that we might encounter. The first one is that the probability of one plan is prominently higher than that of others. This gives the *Planner* a clear idea about the plan the human is executing, and it can directly acquire all the following actions for the robot

Algorithm 1: Proposed HRC system

```
Input plan library $\mathcal{Q}$; motion models $\mathcal{M}$; trajectory prediction models $f^*$; a set of objects $\mathcal{O}$
Init: RobotIsDoing = {}, NewHumanPose = {}, RobotActionBuffer={};
while true do
    NewHumanPose = getValidPoseFromPerception();
    if notEmpty(NewHumanPose) then
        record historical human joint trajectory $h_{1:k}$;
        $m_k$← MotionClassification($h_{1:k}$, $\mathcal{M}$);
        $o_k$ ← TargetObjectEstimation ($m_{1:k}$, $o_{1:k-1}$, $h_{1:k}$);
        $h_{k+1:k+M}$ ← TrajectoryPrediction($h_{1:k}$, $f^*$, $m_k$, $o_k$);
        obtain $a_k = \{m_k, o_k\}$ ;
        generate action trajectory $a_{1:k}$;
        $p(\mathbf{g}|a_{1:k})$, $a_k^{post}$ ← OEDTWPlanInference($\mathcal{Q}$, $a_{1:k}$);
        $\hat{g}^{[1]}$, $\hat{g}^{[2]}$ ← the best and second best plan estimates;
        if $p(\hat{g}^{[1]}|a_{1:k})$ >Threshold then
            RobotActionBuffer← nextActionSequence($\hat{g}^{[1]}$,$a_k^{post}$,$\mathcal{Q}$) ;
        else
            action1 ← nextAction($\hat{g}^{[1]}$,$a_k^{post}$,$\mathcal{Q}$);
            action2 ← nextAction($\hat{g}^{[2]}$,$a_k^{post}$, $\mathcal{Q}$);
            if action1==action2 then
                RobotActionBuffer←action1;
            end
        end
        if notEmpty(RobotActionBuffer) then
            if notEmpty(RobotIsDoing)& RobotActionBuffer{1}!=RobotIsDoing then
                recover what robot is doing;
                RobotIsDoing ← RobotActionBuffer{1};
            end
        end
    end
    Execute(RobotIsDoing, $h_{k+1:k+M}$);
    if ActionExecutionFinished then
        if notEmpty(nextInBuffer) then
            RobotIsDoing ← nextInBuffer;
        else
            RobotIsDoing = {};
        end
    end
end
```

from the plan library. The other case is where there are two or more candidate plans with similar probabilities from the plan recognition algorithms. Under this situation, the *Planner* will look at the two most likely plans, and find out whether the next action of the robot for each plan is the same. If the next action is the same, *Planner* will directly let the robot execute it. Otherwise, the *Planner* will wait and collect the human's information to clear out the confusion.

The *Planner* also takes changes of plan into consideration. If the robot's next action is not consistent with what the robot is doing, it will recover the current action and responsively adjust its action. For example, if the robot is delievering screwdriver to the human, while suddenly the next action becomes bringing the scissors. The robot will put back the screwdriver if it already grabs it, and go to scissors immediately. The pseudo-code for the workflow of our proposed system is presented as Algorithm 1.

VI. Experiments

A. A Desktop Assembly Scenario

We evaluate our proposed framework in a desktop assembly task in industrial settings. The task of the HRC team is to assemble a desktop (desktop assembly example explained in Section II). This task can be decomposed into three unordered subtasks: installing a CPU fan, installing a system fan and taping cables. Each subtask is implemented by only one action sequence. Thus, by a hierarchical decomposition as shown in Fig. 1, there are at most six different plans to finish the task. The robot is designed to assist the human by delivering necessary tools to the human as he/she needs. Definitions of actions for the human and the robot can be found in Section II.

As depicted in the Fig. 1, a predictive robot with an effective plan recognition will recognize the human's plan in the second subtask, and proactively execute the following actions in sequence. For example, if the human is doing the first plan, namely, installing a CPU fan, installing a system fan, and finally taping the cables. We expect that correct plan is inferred when the human is fetching the system fan and then the robot will execute the following actions ("delivering the screwdriver" and "delivering the scissors") in sequence.

B. Experiment Design

1) Hypothesis: We evaluate the effectiveness of the proposed plan recognition and trajectory prediction framework by verifying the following three hypotheses.

- H1: The proposed framework is a safe HRC framework.
- H2: The proposed framework improves the efficiency of the HRC team.
- H3: The performance of the proposed framework is robust to noises or errors caused by some intermediate steps such as motion classification.
- H4: The human subjects are more satisfied with our collaborative robots than with a responsive robot in terms of some criteria.

2) Experiment setup: We test our system on an industrial robot FANUC LR Mate 200iD/7L. A Kinect V2 for windows is placed close to the table on which a robot arm and a human worker do task together. Some necessary tools lie in the tool area and a CPU fan and a system fan are in the part area.

We conducted experiments with human in the loop. Eight human subjects participated in the experiments.

3) Manipulated variables: To evaluate the effectiveness of the proposed framework, we manipulated two controlled variables in our experiments: *plan recognition schemes* and *trajectory prediction schemes*. We define "plan recognition = 0" as no plan recognition, "plan recognition = 1" as recognition ground truths provided by human subjects, and "plan recognition = 2" as the recognition results generated by the proposed algorithm in Section III. When "plan recognition = 0", the robot is completely reactive, meaning that it receives the information of the human action after the human completes the action. The robot only starts to move once it detects the human's actions, and it can only collaborate with human within subtasks. When "plan recognition = 1", the robot has perfect plan knowledge, and it moves based on the ground truths of human's actions and plan. When "plan recognition = 2", the proposed algorithm will let the robot automatically identify the human's actions and infer about the potential plan, so that the human and robot can collaborate across subtasks. In addition, we define "trajectory prediction = 0" as no predictions of human trajectory, and "trajectory prediction = 1" as prediction via our proposed method in Section IV. By manipulating the two variables, we have six groups of experiments. Under each group, every human subject performs the task using any plan for three times. Thus, there will be 24 trials in each experiment group and in total we collect 144 trials for all groups.

4) Dependent measures: To quantify the safety of the proposed framework, we measure the minimum distance between a human subject and the robot during the entire task in each trial. Smaller the minimum distance, the less safe it is for the human. For efficiency, we use a timer to keep track of the task completion time. The timer starts when a human subject starts to move, and ends when the task is finished. To quantify the plan recognition performance, we calculate the plan recognition accuracy and the action recognition accuracy which is intermediate result of the plan recognition module. Plan and action recognition accuracy is the percentage of plan estimates and action estimates that conform to the true values labeled by human subjects. Notice that plan recognition takes place at every time step, and there might be multiple plan labels in the early phase. As long as the estimate is one of the labels, it is regarded as correct. As for the measurement of human's satisfaction with our collaborative robots, we ask the eight human subjects to rate the following statements on Likert scale from 1 (strongly disagree) to 5 (strongly agree), similar to [6]: 1. The robot was collaborative and helped; 2. The robot did the right thing at the tight time; 3. I am satisfied working with the robot; 4. I will work with this robot again in the future.

C. Implementation Details

The motion model was approximated by a one-layer LSTM network. We offline collected 200 trials of "fetching", "receiving", "screwing", and "taping", 50 trials for each class. The hyper-parameter was chosen by cross validation, and we picked 60 to be the number of hidden units.

Human transition model in the module of human trajectory was approximated by a fully connected neural network. We used the data set as described above. We predicted the future trajectory of $1s$ given past trajectory of $1s$. The number of layers was set to 3 and the number of hidden units was set to 40 also by cross validation.

D. Results

H1: Through extensive experiments, the minimum distances between the human subjects and the robot were, respectively, $34.9\pm3cm$ and $36.0\pm2cm$ with "trajectory prediction = 0" (72 trials) and "trajectory prediction = 1" (72 trials). Under both experiment conditions, the minimum distances were within safe distance and no collisions happened. The minimum distances in experiments with "trajectory prediction = 1" were larger than those in the experiments with "trajectory prediction = 0", although not significantly differ[2] (p>0.05). There are two possible reasons: 1) in our scenario, the robot and the human worked in a relatively large space, so predictions of the human trajectory did not make much difference to the robot trajectory planning; 2) human subjects were conservative around the robot, and they intentionally kept a safe distance from the robot.

To show that trajectory prediction module in our framework improves safety, we did additional tests where the human subjects aggressively move towards the robot end effector (We have safety mechanism which immediately stops the robot if contact happens.). 100 trials with trajectory prediction and 100 trials without trajectory prediction were collected. The collision rates were 0/100 and 64/100, respectively. Such results qualitatively showed that the trajectory prediction module can improve safety.

H2: The task completion time for different plan recognition schemes were recorded among trials with the eight human subjects. Without plan recognition ("plan recognition = 0"), the average task completion time is $90.0\pm10.9s$, which is the longest. With our proposed plan recognition algorithm ("plan recognition = 2"), the average task completion time is $64.6\pm10.6s$, which is reduced by 29.1%. Thus, the proposed framework with plan recognition significantly improves ($p < 0.01$) the efficiency of the HRC team compared to the system without the plan recognition. As a matter of fact, our system can achieve similar performance as a system with perfect plan recognition ("plan recognition = 1") with $1.2s$ more average task completion time and larger variance.

H3: This hypothesis can be proved via quantitative results in Table I. One can see that although some motion classification accuracy is low, the plan recognition accuracy still remains high. The Pearson product-moment correlation coefficient for the two variables is $-0.11(p < 0.01)$, which indicates weak correlation. This means that the overall performance of the proposed plan recognition algorithm is not sensitive to the errors in the intermediate LSTM step. This is

[2] We use paired t-test for all the statistical tests.

mainly benefiting from the Bayesian inference step and the dynamic time warping step. These two steps serve as a low pass filter, eliminating the wrong motion estimates. Besides, the plan is actually estimated by the nearest neighbor in DTW step, and the six plans as action sequences lie sparsely in an increasingly high dimensional space, and they get farther away from each other over time. As long as the estimates do not deviate from the true point too much, the plan estimate should be correct.

TABLE I: Quantitative experimental results for recognition

Subjects	**MC accuracy(%)**	**PR accuracy (%)**
Subject 1 (6 trials)	85.3 ± 1.3	97.6 ± 0.3
Subject 2 (6 trials)	69.6 ± 15	98.5 ± 0.5
Subject 3 (6 trials)	81.4 ± 1.1	97.4 ± 1.2
Subject 4 (6 trials)	87.9 ± 7.0	97.6 ± 1.1
Subject 5 (6 trials)	79.4 ± 6.5	96.4 ± 1.8
Subject 6 (6 trials)	80.7 ± 12.0	97.9 ± 0.1
Subject 7 (6 trials)	84.8 ± 9.5	97.6 ± 0.6
Subject 8 (6 trials)	85.8 ± 8.8	90.5 ± 0.3

To further validate the robustness brought by the DTW step, we assumed that the target object detection was perfect and did simulations by varying the motion classification (MC) performance and then tested the plan recognition (PR) accuracy. First, we obtained the true positive rates for each motion throughout all the experiments with "plan recognition = 2": 83.4% for "screwing", 64.2% for "fetching", 59.1% for "receiving", and 84.2% for "taping". Then we varied each true positive rate by $\delta(\%)$, which took values of $0, -5, -10, -15, -20, -30, -40,$ and -45. Based on these sets of true positive rates, we simulated 15 trials of action sequences for each of the six plans in the desktop assembly task, and so we had 90 trials for each δ. As we can see in the Fig. 4, when δ is equal to -30 (true positive rate for each motion is $53.4\%, 24.2\%, 29.1\%, 54.2\%$) and the overall motion classification is 30%, the plan recognition accuracy remains higher than 85%, which shows robustness of our plan recognition to the motion accuracy.

H4: Fig. 5 shows the comparison of human subjects' ratings for the six types of robots on four criteria mentioned above. Human subjects rated the robot with our proposed plan recognition algorithm ("plan recognition = 2") significantly higher ($p < 0.01$) than the robot without plan recognition ("plan recognition = 0") on all four criteria. Between robots with ground truths of plan recognition ("plan recognition = 1") and robots with our plan recognition algorithm ("plan recognition = 2"), there is no significant difference ($p > 0.05$) on all the criteria except for the criteria "The robot did the right things at the right time" ($p = 0.04$). Furthermore, there is also no significant difference ($p > 0.05$) between the robots with trajectory prediction ("trajectory prediction" = 1) and the robots without trajectory prediction ("trajectory prediction" = 0). This might be because the trajectory prediction is too short to influence the human's feedback. Recalling the fact that "plan recognition" has significant influence, we can see that human care more about efficient plan recognition.

TABLE II: The result table for the three tasks

Tasks	**Precision**		**Recall**		**$F_{0.5}$**	
	ours	MEMM	ours	MEMM	ours	MEMM
Drinking water	97.0	87.9	95.4	80.8	96.7	86.4
Cooking (stirring)	88.9	65.5	98.4	43.9	90.6	59.7
Opening pill container	84.3	86.4	87.0	58.0	84.4	78.7

Aside: The sensitivity of threashold is not obvious. We designed a new experiment by varying the threshold value in Algorithm 1 on the experiment data we collected. It was found that the plan recognition results remained the same when the threshold was dropped from 0.70 to 0.58, where 0.70 is the threshold we used in other experimental results.

VII. Extension to Multiple Tasks

Our plan recognition algorithm also works to distinguish different tasks. We test our algorithm on three different tasks in the CAD-60 Cornell Activity Datasets [14]: cooking (stir), opening a pill container and drinking water. Table II shows the comparison of the results in the "new person" setting using our algorithm and the two-layered maximum entropy Markov model (MEMM) method in [14]. We can see that our plan recognition algorithm can also achieve very high accuracy compared to the approach in [14]. This verifies our claim that exploiting the rich object information can help improve the task/plan recognition performance.

In addition, compared to end-to-end leanring, our algorithm advantages in two aspects: 1) the learning process is easier, since a hierarchical combination of motion classification and target estimation reduces the dimension of the classification problem; 2) the learning pipeline is more interpretable and predictable. As a cost, however, the proposed method requires stronger prior knowledge, i.e., all possible plans of the new task should be predefined offline, which might be hard when the task is complicated.

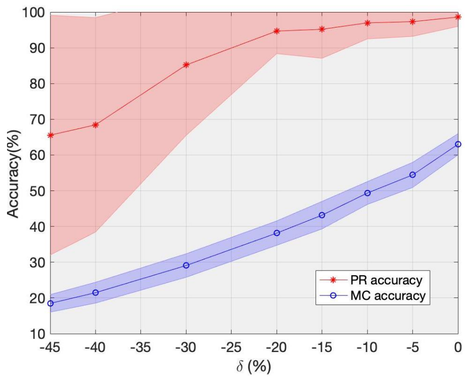


Fig. 4: Simulations for plan recognition accuracy when reducing the motion classification accuracy.

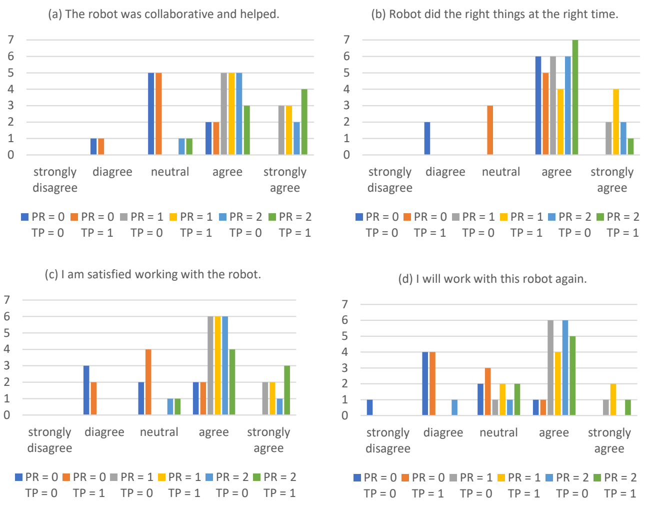


Fig. 5: Human subjects‘ ratings for six types of robots on four different criteria. PR is short for plan recognition and TP is short for trajectory prediction.

VIII. Conclusion

In this paper, we proposed an integrated framework for human-robot collaboration, including both the plan recognition and trajectory prediction. By explicitly leveraging the hierarchical relationships among plans, actions and trajectories, we designed a robust plan recognition algorithm based on neural networks and Bayesian inference. Experiments with human in the loop were conducted on an industrial assembly task. The results showed that with our proposed framework, the efficiency and safety of the human-robot collaboration can be improved. The average task completion time was reduced by 29.1%. Moreover, the proposed plan recognition algorithm was robust and reliable. Correct plan recognition was achieved even when the motion labels via neural networks are of low accuracy. The trajectory prediction module also enhanced the safety of the human by keeping a safe distance between the human and the robot, which is shown in our experiment video.

We verified the effectiveness of the proposed algorithms on both an designed computer-assembly experiment and the CAD60 dataset, with comparison to the MEMM approach. More extensive comparison studies will be explored in the future to fully investigate the advantages of the proposed work, with integration with different human-robot collaboration frameworks.

References

[1] A. Thomaz, G. Hoffman, M. Cakmak *et al.*, “Computational human-robot interaction,” *Foundations and Trends® in Robotics*, vol. 4, no. 2-3, pp. 105–223, 2016.

[2] S. Devin, A. Clodic, and R. Alami, “About decisions during human-robot shared plan achievement: Who should act and how?” in *International Conference on Social Robotics*. Springer, 2017, pp. 453–463.

[3] C. Pérez-D’Arpino and J. A. Shah, “Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification,” in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 6175–6182.

[4] S. Pellegrinelli, H. Admoni, S. Javdani, and S. Srinivasa, “Human-robot shared workspace collaboration via hindsight optimization,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 831–838.

[5] C. Liu, J. B. Hamrick, J. F. Fisac, A. D. Dragan, J. K. Hedrick, S. S. Sastry, and T. L. Griffiths, “Goal inference improves objective and perceived performance in human-robot collaboration,” in *Proceedings of the 2016 international conference on autonomous agents & multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2016, pp. 940–948.

[6] H. S. Koppula, A. Jain, and A. Saxena, “Anticipatory planning for human-robot teams,” in *Experimental Robotics*. Springer, 2016, pp. 453–470.

[7] M. Kohler *et al.*, *Using the Kalman filter to track human interactive motion: modelling and initialization of the Kalman filter for translational motion*. Dekanat Informatik, Univ., 1997.

[8] A. Bruce and G. Gordon, “Better motion prediction for people-tracking,” in *Proc. of the Int. Conf. on Robotics & Automation (ICRA), Barcelona, Spain*, 2004.

[9] P. Ghosh, J. Song, E. Aksan, and O. Hilliges, “Learning human motion models for long-term predictions,” in *3D Vision (3DV), 2017 International Conference on*. IEEE, 2017, pp. 458–466.

[10] L. Sun, W. Zhan, and M. Tomizuka, “Probabilistic Prediction of Interactive Driving Behavior via Hierarchical Inverse Reinforcement Learning,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov. 2018, pp. 2111–2117.

[11] L. Sun, W. Zhan, Y. Hu, and M. Tomizuka, “Interpretable modelling of driving behaviors in interactive driving scenarios based on cumulative prospect theory,” in *Proceedings of the 2019 IEEE Intelligent Transporation Systems (ITSC2019)*, Oct. 2019, pp. 4329–4335.

[12] Y. Cheng, W. Zhao, C. Liu, and M. Tomizuka, “Human motion prediction using semi-adaptable neural networks,” in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 4884–4890.

[13] V. V. Unhelkar, P. A. Lasota, Q. Tyroller, R.-D. Buhai, L. Marceau, B. Deml, and J. A. Shah, “Human-aware robotic assistant for collaborative assembly: Integrating human motion prediction with planning in time,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2394–2401, 2018.

[14] J. Sung, C. Ponce, B. Selman, and A. Saxena, “Human activity detection from rgbd images,” in *Workshops at the twenty-fifth AAAI conference on artificial intelligence*, 2011.

[15] S. Mukherjee, L. Anvitha, and T. M. Lahari, “Human activity recognition in rgb-d videos by dynamic images,” *arXiv preprint arXiv:1807.02947*, 2018.

[16] C. Zhang and Y. Tian, “Rgb-d camera-based daily living activity recognition,” *Journal of computer vision and image processing*, vol. 2, no. 4, p. 12, 2012.

[17] M. Fiore, A. Clodic, and R. Alami, “On planning and task achievement modalities for human-robot collaboration,” in *Experimental Robotics*. Springer, 2016, pp. 293–306.

[18] S. Nikolaidis, A. Kuznetsov, D. Hsu, and S. Srinivasa, “Formalizing human-robot mutual adaptation: A bounded memory model,” in *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*. IEEE Press, 2016, pp. 75–82.

[19] S. Nikolaidis and J. Shah, “Human-robot cross-training: computational formulation, modeling and evaluation of a human team training strategy,” in *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*. IEEE Press, 2013, pp. 33–40.

[20] L. Sun, W. Zhan, M. Tomizuka, and A. D. Dragan, “Courteous autonomous cars,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 663–670.

[21] C. Liu and M. Tomizuka, “Robot safe interaction system for intelligent industrial co-robots,” *arXiv preprint arXiv:1808.03983*, 2018.

[22] D. Das and C. S. G. Lee, “Cross-scene trajectory level intention inference using gaussian process regression and naive registration,” 2018.

[23] C. L. Baker, R. Saxe, and J. B. Tenenbaum, “Action understanding as inverse planning,” *Cognition*, vol. 113, no. 3, pp. 329–349, 2009.

[24] J. N. Kapur and H. K. Kesavan, “Entropy optimization principles and their applications,” in *Entropy and energy dissipation in water resources*. Springer, 1992, pp. 3–20.

[25] P. Tormene, T. Giorgino, S. Quaglini, and M. Stefanelli, “Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation,” *Artificial intelligence in medicine*, vol. 45, no. 1, pp. 11–34, 2009.