Towards Better Human Robot Collaboration with Robust Plan Recognition and Trajectory Prediction

Yujiao Cheng, Liting Sun, and Masayoshi Tomizuka

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. However, due to the stochastic and time-varying nature of human collaborators, it is challenging for the robot to efficiently and accurately identify the plan of human and respond in a safe manner. To address this challenge, we propose an integrated human robot collaboration framework in this paper which includes both plan recognition and trajectory prediction. Such a framework enables the robot to perceive, predict and adapt its actions to the human’s plan and intelligently avoid collisions with the human based on the predicted human trajectory. Moreover, by explicitly leveraging the hierarchical relationship between the plan and trajectories, more robust plan recognition performance can be achieved. Experiments are conducted on an industrial robot to verify the proposed framework, which shows that our proposed framework can not only assure safe HRC, but also improve the time efficiency of the HRC team, and the plan recognition module is not sensitive to noises.

I. INTRODUCTION

As the emphasis of manufacturing is shifting from mass production to mass customization [1], 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. As shown in Fig. [1] the idea of HRC is to let robots work safely and collaboratively with humans in a shared space. To achieve this, it is required that the robot should be able to perceive, understand and anticipate the behavior of the human workers, and adapt itself to collaborate with them in an intelligent manner [2].

Depending on the intelligence of collaborative robots (Co-Robots), the collaboration between humans and robots can be categorized into three different levels: from 1) low-level collision avoidance, to 2) middle-level efficient cooperation with prediction and recognition for human’s trajectories, intentions and plans, to 3) high-level collaboration mode selection and task assignments. Many researches have been conducted for all these three levels. For example, for the first category, [3] [4] [5] regarded humans as moving obstacles in the environments, the goals, the plans and the actions. The estimated mental states were then utilized to better execute human-robot shared plans. Huang et al. [8] proposed a robot system that monitors its user’s gaze, predicts his or her task intent based on observed gaze patterns, and performs anticipatory actions according to its predictions. As for the research of the third category (high-level collaboration mode selection and task assignments), [9] [10] studied the task assignments in peer-to-peer human robot interaction where humans and robots work as partners.

We focus on the middle level of HRC in this paper. To be more specific, we aim to improve the efficiency of human-robot collaboration while assuring safety via reliable plan recognition and trajectory prediction. To achieve this, an integrated HRC framework is proposed, which enables the robot to perceive the human’s actions, infer the human’s plans and adapt to the human’s actions to achieve efficient and safe collaboration. 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 mind. By using our proposed plan recognition method, the robot can quickly recognize the human’s 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. We combine a long short-term memory (LSTM) network with algorithms based on Bayesian inference instead of completely relying on the output of the network. 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 to improve the robustness of the algorithm, but also to reduce the dimension of the problem where the third advantage of our framework comes from. The proposed framework has good generalization ability with less data. All these advantages of the proposed framework have been verified via experiments on an industrial assembly settings in Section VI.

The remainder of this paper is organized as follows. In section II an integrated HRC framework is proposed. In section III we present the detailed algorithm of the plan recognition part, followed by a description of a trajectory prediction method in section IV. Section V describes the mechanism of the planner. We show the experiments in section VI and conclude the paper in section VII.

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 object to act on (the target object), i.e., action=\{motion, object\}. For example, we can define “action l=\{Fetching, a screwdriver\}”, “action 2=\{Taping, 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.

In HRC systems, humans and robots collaborate to finish a given task. However, as defined above, for an arbitrary task, there might be a variety of plans to conduct it. For example, Fig. 2 shows an example of the hierarchical decomposition of a task. The task is composed of three subtasks, and each of them has a unique action order. Thus, the permutation of the three subtasks generates totally six different plans. Within each plan, there are infinite many (theoretically) trajectories to execute. Moreover, human plans are in nature time-inconsistent and stochastic, i.e., human collaborators might change plans in the middle of a task, and different human collaborators might have different preferences over plans. Hence, efficient and accurate recognition of human plans is necessary and of great importance to improve the efficiency and smoothness of HRC. For instance, as shown in Fig. 2, without plan recognition, the robot can only acquire its next actions after the human finishes some key actions (the “reactive” robot in Fig. 2). While with a plan recognition, the robot (the “predictive” robot in Fig. 2) can execute its next actions in advance to boost the efficiency of the collaboration.

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 assure efficient and seamless collaboration.

To address these two challenges, we propose an integrated HRC framework, the architecture of which is shown in Fig. 3. It includes both offline database and online modules such as a perception module (sensors and perception algorithms), a plan recognition module, a trajectory prediction module, a planner, a motion control module and the actuators (the robot). The goal of each online module is given below.

Perception Module. Perception module takes visual information as inputs and outputs the 3D positions of objects as well as the 3D human poses. Many algorithms based on deep learning techniques can be utilized to achieve this. Since this module is not the focus of this paper, we omit the details here.
Plan Recognition Module. This module is a key module in our proposed framework. It aims to identify the action being executed by the human and infer humans plan by observing the trajectories of their key joints. Given the defined hierarchical relationships, the plan recognition module includes the following key steps: 1) identification of motion labels, 2) inference of target object, 3) inference of possible plans based on the sequence of actions obtained in step 1) and 2), and 4) posterior action estimation update based on the plan information from step 3). The offline constructed database will be utilized in all four steps, and the detailed algorithms for each step are provided in later sections. The recognized plan and the posterior action estimate will be sent to the planner module to let the robot adapt its behaviors.

Planner Module. The planner module assigns the next action (a motion-object pair) to the robot based on the current states and the recognized plan and the current action of the human. The action command from the planner is sent to the motion control module.

Trajectory Prediction Module. This 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 explicitly take advantage of the hierarchical relationships among plans, actions and trajectories. Therefore, we design the trajectory prediction module to leverage two inputs: 1) the human pose estimates from perception module and 2) the action labels from the plan recognition module. The introduction of the plan and motion information can help achieve better trajectory prediction results. Moreover, we also bring in online adaptation to address the challenges regarding the time-varying characteristic of human trajectories. Detailed algorithms and experimental results are provided in later sections.

Motion Control Module. The motion control module includes two controllers: an efficiency controller and a safety controller, as in [11]. 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 to guarantee 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. 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 so that better plan recognition can be achieved. As mentioned in Section II, the four steps of plan recognition are: motion classification, target object estimation, plan inference and posterior action estimation update.

The proposed plan recognition algorithm have 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 (step 4) based on 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 [12]. Hence, we design an LSTM recurrent neural network for motion classification. The structure of the LSTM network is depicted in Fig. 4. The input data is the human pose from the Perception module. To be more specific, in an assembly task, the input vector at time step $k$ is $x_k = \{w_k, h_k\}$, where $w_k$ is the wrist position in the world frame at time step $k$ and $h_k$ 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 offline-trained using the “Motion Model” database in Fig. 5.

---

**Fig. 3:** The architecture of the proposed integrated HRC framework

**Fig. 4:** The LSTM network structure for motion classification
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. [13] [14].

Let \( o_k \) be an object at time step \( k \), \( O \) 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:

\[
P(o_k|h_{1:k}, m_{1:k}) \propto P(m_k|o_k, h_{k-1}, m_{k-1}) \cdot P(h_k|o_k, h_{k-1}, m_k) \\
\sum_{o_{k-1} \in O} P(o_k|h_{k-1}, m_{k-1}, o_{k-1}) \cdot P(o_{k-1}|h_{k-1}, m_{k-1})
\]

We compute the \( P(h_k|o_k, h_{k-1}, m_k) \) with an assumption that humans are optimizing some value function as [14] 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 [15]. We adopt the open-end dynamic time warping (OE-DTW) algorithm [16] 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 \( \phi_t \) and \( \psi_t \), \( t = 1, 2, ..., T \), that satisfy the following constraints:

- Boundary condition: \( \phi_1 = 1, \psi_1 = 1 \) and \( \phi_T = N, \psi_T = M \)
- Monotonic conditions: \( \phi_t \leq \phi_{t-1} \) and \( \psi_t \leq \psi_{t-1} \)
- 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 [16].

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

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

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

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

D. Posterior Action Correction

As we obtain the posterior estimate of the plan \( g^* \), the best matched reference sequence \( R^* \) 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 the safety constraint. Nevertheless, human trajectory is inherently difficult to predict due to the nonlinearity and stochasticity in the human behavior. In addition, individual differences are also prominent. Prediction models that work for one person may not be applicable
to another. To cope with this problem, we introduce an adaptable neural network method.

The transition model of human joint motion at time step $k$ is formulated as

$$x(k+1) = f^*(x^*(k), m, q) + w_k,$$

where $x(k+1)$ is a vector denoting the human’s future $M$-step trajectory of the joint, also noted as $h_{k+1:k+M}$ consistent with previous sections. $x^*(k)$ denotes the human’s past $N$-step trajectory of the joint, which is also $h_{k-N+1:k}$. Horizon $M, N \in \mathcal{N}$ can be determined by cross validation or by experience. Motion label $m \in \mathcal{N}$ is obtained by motion classification in Plan Recognition module. $q$ is the position vector of the target object. $w_k$ is a zero-mean white Gaussian noise.

This model can be approximated by a feedforward neural network, and the output layer can be adapted online using recursive least square parameter adaptation algorithm (RLS-PAA). By some derivations, uncertainty level of the predictions can also be computed. Both the predicted trajectory and uncertainty level will then be sent to safety controller to generate safety constraint.

For more details of this approach, one can refer to our previous work [17].

V. THE PLANNER

In this section, we will present the Planner module and the workflow of the proposed framework. As shown in Fig. 4, 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. For example, if the robot’s next action is to bring the screwdriver to the human, the Planner module will set a sequence of the goal states in the Motion control module as "end effector to be at the position where the screwdriver is located", "grasp the screwdriver", "end effector to a position near the human’s hands". In the meanwhile, the predicted trajectory of human joints will also be sent to Motion control module to generate safety constraints.

Since the plan recognition results are probabilistic, we need to design a decision-making mechanism in the Planner module. There are two cases that we might encounter. The first one is where there is one plan whose probability is prominently higher than the others. This gives the Planner a clear idea about the plan that the human is following, and it can directly acquire all the next actions for the robot from the plan library. The other case is where there are two or more candidate plans with similar probabilities from the plan recognition algorithms. We define this as a "confusion" phase for the planner. Under this situation, the Planner will look at the two most likely plans, and find the next action of the robot for each of the two plans and compare the two next actions of the robot. If next action of the robot is the same, then the 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 bringing
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 Benchmark Human Robot Collaboration Scenario

We evaluate our proposed framework on a benchmark scenario in industrial assembly. In this benchmark scenario, the task of the HRC team is to assemble a desktop, as depicted in Fig. 5. This task can be decomposed into three un-ordered subtasks: installing a CPU fan, installing a system fan and taping a bunch of cables. Each subtask is implemented by one action sequence. Thus, by a hierarchical decomposition as shown in Fig. 2 there are at most six different plans to finish the task. The robot is defined to assist the human by bringing necessary tools to the human as he/she needs. For example, when the human is about to install the CPU fan or the system fan, he/she will need a specific screwdriver in the tool area, and the robot should be able to bring it to him/her in a timely and safe manner. When the human is taping the cables, the robot should know that the human needs the scissors in the near future and bring it to the human.

As depicted in the Fig. 2, a predictive robot with an effective plan recognition will recognize the human’s plan in the second subtask, and proactively execute the next actions in sequence. For example, if the human is doing the first plan, namely, installing a CPU fan, installing a system fan, and finally tapping the cables. We expect that correct plan is inferred when the human is fetching the system fan (we call this action as the key action), and then the robot will execute the next actions (bringing the screwdriver and bringing 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 assures safe HRC.
   • H2: The proposed framework improves the efficiency of the HRC team.
   • H3: The performance of the proposed framework is not sensitive to noises or errors caused by the motion classification step via LSTM.

2) Experiment setup: We test our system on an industrial robot FANUC LR Mate 200iD/7L. As shown in Fig. 5, the vision sensor we use is a Kinect V2 for windows, placed closed to the table on which a robot and a human 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. Six human subjects are invited in the experiments.

3) Manipulated variables: To evaluate the effectiveness of the proposed framework, we manipulated one controlled variable in our experiments: plan recognition schemes. We define “plan recognition = 0” as no plan recognition, “plan recognition = 1” as recognition via a human expert, and “plan recognition = 2” as recognition via the proposed algorithm in Section III. When “plan recognition = 0”, the robot is completely reactive, meaning that it receives the information of the action of human after the completion of the action, and then it starts to help according to the human’s need. When “plan recognition = 1”, a human expert observes the human worker’s action and manually inputs that information to the robot once the expert recognizes the action and plan. When “plan recognition = 2”, the proposed algorithm will be running to let the robot automatically identify the human worker’ action and infer about the potential plan.

4) Dependent measures: To quantify the safety of the proposed framework, we measure the closest distance between a subject human worker and the robot during the entire task.
For efficiency, we use a timer to keep track of the task completion time. The Timer starts when a subject human worker starts to move, and ends when the task is finished.

To quantify the plan recognition performance, we calculate the plan recognition accuracy and delay index of the proposed algorithm. Plan recognition accuracy is the percentage of plan estimates that conform to the true values labeled by human experts. Note that there can be more than one plan labels at a time step in the early phase, and as long as the estimate is one of the labels, it is regarded as correct. The delay index is defined as $(t - t_h)/T_{KeyAction}$, where $t$ and $t_h$ are the times when our proposed method and the human expert succeed to recognize the true plan in the human subject’s brain, and $T_{KeyAction}$ is the duration of the key action at which that the plan is recognized. One example of key action is "fetching the system fan" in Section VI-A.

### C. Results

**a) H1: The proposed framework assures safe HRC:** Through extensive experiments, the minimum distance between the human subjects and the robot for different plans of the task is greater than 30cm, a threshold safety distance specified in our system. Hence, it is verified that the proposed framework can guarantee the safety of HRC. An example of the trajectory is visualized in Fig. 7. It shows that given the initial and goal states (the red cross markers in Fig. 7), the robot detours to avoid collisions with the human (the purple trajectory in the figure) instead of directly going to the goal states along the shortest kinematically feasible trajectory depicted by the dotted red line in the figure. More examples are provided in the supplementary video.

**b) H2: The proposed framework improves the efficiency of the HRC team:** The task completion time for different plan recognition schemes are recorded among trials with the six human subjects. Results are shown in Fig. 6. It can be seen that without plan recognition ("plan recognition = 0"), the average task completion time is longer. With our proposed plan recognition algorithm ("plan recognition = 2") in Section III, the average task completion time is significantly reduced. As a matter of fact, the proposed algorithm can achieve similar performance as a human expert ("plan recognition = 1") in terms of efficiency.

**c) H3: The performance of the proposed framework is not sensitive to noises or errors caused by the motion classification step via LSTM:** This hypothesis can be proved via the quantitative results in Table I. The "Motion Classification Accuracy" calculates the ratio of correctly identified motions, and "Plan Recognition Accuracy" is the percentage of plan estimates that conform to the labels of human experts. One can see that although the motion classification accuracy is not very high for some subjects, the plan recognition accuracy can still remain high. This means that the overall performance of the proposed plan recognition algorithm is not sensitive to the errors in the intermediate LSTM step, benefiting from the Bayesian inference steps based on the hierarchical relationships defined in Fig. 2.

Besides the plan recognition accuracy, we also have calculated the delay index. Delay index is the ratio between delayed time for our plan recognition, compared with the human expert, and the duration of the key action. The results are given in Table I and Fig. 8. We can see that the plan recognition delay indexes also remain low even though the motion classification accuracies are low in some trials. Hence, both results from the plan recognition accuracy and the plan recognition delay index show that our proposed plan recognition algorithm is robust to errors in the intermediate LSTM steps.

### VII. 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 data-efficient and robust plan recognition algorithm based on neural networks and Bayesian inference. Experiments with human in the loop were conducted focusing on an industrial assembly task. The results showed that with our proposed framework, the efficiency and safety of the
human robot collaboration have been greatly improved. For efficiency, the average task completion time was significantly reduced, about the same level as an human expert. 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. For safety, the trajectory prediction module guaranteed the safety of the human by keeping a safe distance between the human and the robot, which is also shown in our experiment video.

Fig. 8: The relationship between the motion classification accuracy and the plan recognition delay index for each trial.

REFERENCES

[1] B. J. Pine, Mass customization: the new frontier in business competition. Harvard Business Press, 1999.
[2] 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.
[3] F. Flacco, T. Krger, A. De Luca, and O. Khatib, “A depth space approach to human-robot collision avoidance,” in 2012 IEEE International Conference on Robotics and Automation. IEEE, 2012, pp. 338–345.
[4] L. Balan and G. M. Bone, “Real-time 3d collision avoidance method for safe human and robot coexistence,” in 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2006, pp. 276–282.
[5] D. M. Ebert and D. D. Henrich, “Safe human-robot-cooperation: Image-based collision detection for industrial robots,” in IEEE/RSJ international conference on intelligent robots and systems, vol. 2. IEEE, 2002, pp. 1826–1831.
[6] O. C. Schrempf, U. D. Hanebeck, A. J. Schmid, and H. Worn, “A novel approach to proactive human-robot cooperation,” in Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on. IEEE, 2005, pp. 555–560.
[7] S. Devin and R. Alami, “An implemented theory of mind to improve human-robot shared plans execution,” in Human-Robot Interaction (HRI), 2016 11th ACM/IEEE International Conference on. IEEE, 2016, pp. 319–326.
[8] C.-M. Huang and B. Mutlu, “Anticipatory robot control for efficient human-robot collaboration,” in The Eleventh ACM/IEEE International Conference on Human Robot Interaction. IEEE Press, 2016, pp. 83–90.
[9] T. Fong, I. Nourbakhsh, C. Kunz, L. Fluckiger, J. Schreiner, R. Ambrose, R. Burridge, R. Simmons, L. Hiatt, A. Schultz et al., “The peer-to-peer human-robot interaction project,” in Space 2005, 2005, p. 6750.
[10] 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.
[11] C. Liu and M. Tomizuka, “Robot safe interaction system for intelligent industrial co-robots,” arXiv preprint arXiv:1808.03983, 2018.
[12] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, “Global context-aware attention lstm networks for 3d action recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1647–1656.
[13] D. Das and C. Lee, “Cross-scene trajectory level intention inference using gaussian process regression and naive registration.” 2018.
[14] C. L. Baker, R. Saxe, and J. B. Tenenbaum, “Action understanding as inverse planning,” Cognition, vol. 113, no. 3, pp. 329–349, 2009.
[15] 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.
[16] 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.
[17] Y. Cheng, W. Zhao, C. Liu, and M. Tomizuka, “Human motion prediction using adaptable neural networks,” arXiv preprint arXiv:1810.00781, 2018.