Efficient and Collision-Free Human-Robot Collaboration Based on Intention and Trajectory Prediction

Jianzhi Lyu, Philipp Ruppel, Norman Hendrich, Shuang Li, Michael Görner and Jianwei Zhang, *Member, IEEE,*

***Abstract*—Human-Robot collaboration (HRC) is an important topic for manufacturing and household robotics. It is very challenging to ensure both efficiency and safety in HRC. This paper presents an HRC pipeline that generates efficient and collision-free robot trajectories based on predictions of the human arm and hand (AH) motions. We train a recurrent neural network for AH trajectory prediction based on observed initial trajectory segments. To increase the accuracy of target estimation at an early stage, the observed and the predicted hand palm trajectory are combined to predict the current AH motion target using Gaussian Mixture Models (GMMs). An optimization-based trajectory generation algorithm is proposed to ensure the safety of the human while collaborating with the robot. The proposed system is validated in a shared-workspace scenario with human pick-and-place motions. The task can be safely and efficiently completed. The results demonstrate that our proposed pipeline can predict the human AH trajectory and estimate the motion target intended by the human accurately and early.**

***Index Terms*—Human-robot collaboration, Collision avoidance, Motion prediction, Intention recognition.**

I. Introduction

ROBOTS are powerful and fast, while humans are intelligent and can carry out dexterous manipulation tasks that may be hard for robots. Human-robot collaboration is increasingly used in order to improve work efficiency and flexibility. However, it is very challenging to ensure the safety of the human and the efficiency of the robot at the same time. In this paper, we consider a scenario shown in Fig. 1, where a human and robot share a narrow workspace. Physical interactions like compliance control in [1] are not considered here. But the robot should work together with the human. To improve the joint assembly tasks' efficiency and ensure human safety in the shared workspace, the robot needs to be able to predict the human AH trajectory and infer the human's target position in a short time horizon.

We model human motions using four joints, i.e., shoulder, elbow, wrist, and palm. Some work has been proposed to predict human trajectories in similar scenarios, such as [2]–[4]. But these works just consider one or two joints, like the wrist and elbow, which is not enough to make sure that the robot can avoid the human AH, especially in this narrow workspace.

This work was partially supported by the German Science Foundation (DFG) in the project Crossmodal Learning, TRR-169, and the CSC (China Scholarship Council). (*Corresponding author: Jianzhi Lyu.*)

All authors are with the Group TAMS, Department of Informatics, Universität Hamburg, 22527 Hamburg, Germany (e-mail: lyu@informatik.uni-hamburg.de).

Manuscript received xxx; revised xxx

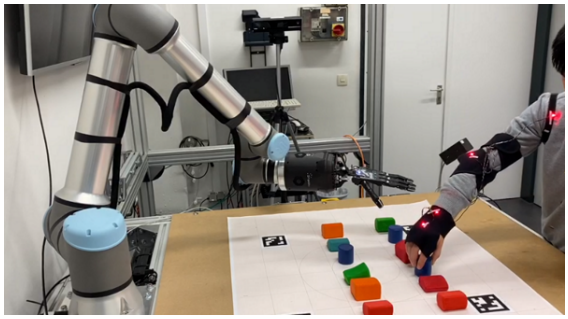

Fig. 1. Side view of the human-robot pick-and-place platform used for our experiments. In the shared workspace, the human puts objects to one of twelve target positions. The robot observes the current AH trajectory and predicts both the short-horizon human motion and the human reaching target position. The robot then generates collision-free and goal-oriented trajectories online to collaborate with the human for the assembly task.

In [5], an adaptive method is used to predict the human hand trajectory. However, it is hard to use in our task because the problem dimension is too high to adjust the weights online. In general, creating an accurate dynamic model for human AH motion prediction is difficult, especially for different persons. So the state-of-the-art, e.g. [6], [7], are based on data-driven models. Inspired by [6], we also adopt a position-velocity encoder-decoder neural network for AH trajectory prediction.

In fact, we could train a neural network to predict the short-term AH trajectory and estimate the intended final target position at the same time. However, training such a multi-task neural network would be challenging because of more parameters to tune, and it would also be hard to take semantics (like a set of known target positions) into account for human intention prediction. Instead, probabilistic methods have been preferred for intended target inference or motion regression [8], [9] and they can generalize well to new scenarios. The computational load of these methods is quite low when the number of possible target positions is limited (a total of 12 targets in this paper, as shown in Fig. 2), so they are efficient enough for fast motion prediction. Besides, the probabilistic methods could be trained quickly in an unsupervised way. Whereas the above work only estimated human intentions based on the observed trajectory and is not suitable for our task. In our scenario, the targets are very close to each other (with 10 cm in between), and the initial parts of the trajectories (about 50%) are very similar, as shown in Fig. 3a and 3b. To increase the target estimation accuracy during the early stage of the reaching motion, we propose using both observed and

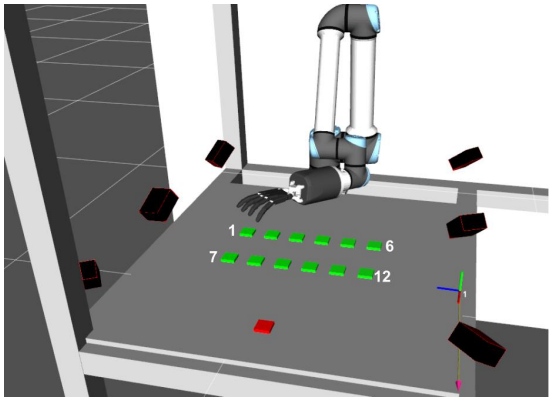

Fig. 2. Placement of the target and initial positions. The green markers indicate the target positions (1-12), and the red marker at the front represents the human hand's initial (and rest) position. The black boxes show the cameras of the motion-capture system used to track human motion.

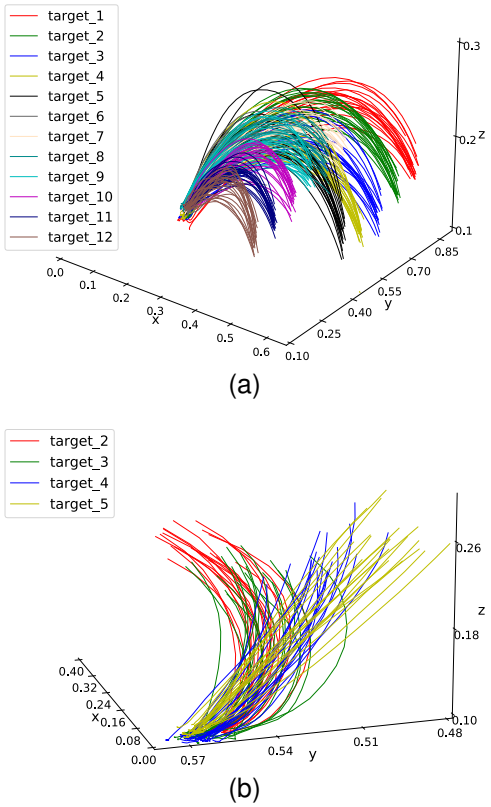

Fig. 3. The units of axes are meters. **(a)** Recorded human reaching trajectories (human palm) from the rest position to the twelve target positions. **(b)** Close view of the first 40% of a few reaching trajectories for target positions 2-5, demonstrating the initial overlap between the trajectories.

predicted palm trajectories based on GMMs.

Safe robot trajectories were generated online by solving two optimization problems in [10], [11], but under the assumption that the AH trajectory prediction was known already. Based on the predicted AH trajectory from our proposed motion prediction module, we can efficiently generate a safe robot trajectory by solving only one optimization problem with fewer objective functions. To significantly reduce the number of geometric constraints in the trajectory optimization problem, we employ capsules to model AH and robot links instead of a large number of spheres as in previous work.

To summarize, the key contributions of this work are as follows:

- We propose an HRC pipeline that combines high-level and multi-joint trajectory prediction and intended target estimation with low-level online collision-free trajectory generation for HRC tasks.
- Not only the observed trajectories but also predicted palm trajectories are used to estimate the final intended AH target, thus increasing the target estimation accuracy during the early motion stage by unsupervised learning.
- We evaluate the proposed HRC pipeline by real physical experiments. The results show that the robot can generate goal-oriented and collision-free trajectories to improve the efficiency and safety of HRC.

II. RELATED WORK

A. Human Trajectory Prediction

Different methods have been proposed for human trajectory prediction. Human joint-space trajectories are predicted based on dynamical movement primitives (DMP) and then used to predict human joint torques for intention estimation during walking [12]. Another category of algorithms for AH motion prediction is based on Inverse Optimal Control (IOC), which tries to approximate a cost function explaining the observed behavior, e.g., [13]. However, with IOC the goal information needs to be known first, which is not possible for the task we are interested in. Other work predicts human motions with explicitly defined dynamic equations derived from physical theory, such as [14]–[16]. But it is hard to model human dynamics, and the model-based methods usually only work well for a very short time horizon.

There are also some pattern-based approaches. They can learn complex dynamic models from datasets based on all kinds of approximation methods (e.g., neural networks, Hidden Markov Models, GMMs). Luo et al. used the GMMs to model the human AH trajectory in [17]. This unsupervised method can generalize to new persons by dynamically updating or generating new models. Wang et al. proposed a position-velocity recurrent encoder-decoder neural network (PVRED) [6]. A velocity connection is added to the input of the long short-term memory (LSTM), and their results show that this method can achieve a better performance than previous results. We revised this model to predict the AH trajectory in our scenario.

B. Human Intention Estimation

Human gaze, gestures, electroencephalography (EEG), electromyography (EMG), etc., could be used for human intention estimation [18]–[20]. Here we focus on the algorithms that make use of human reaching motions. Arpino and Shah predicted the reaching target by time series classification in [21]. They encoded each time step as a multivariate Gaussian distribution and calculated the class posterior probability with the observed trajectory. The result shows that they can achieve a rather accurate target prediction. A similar idea has been

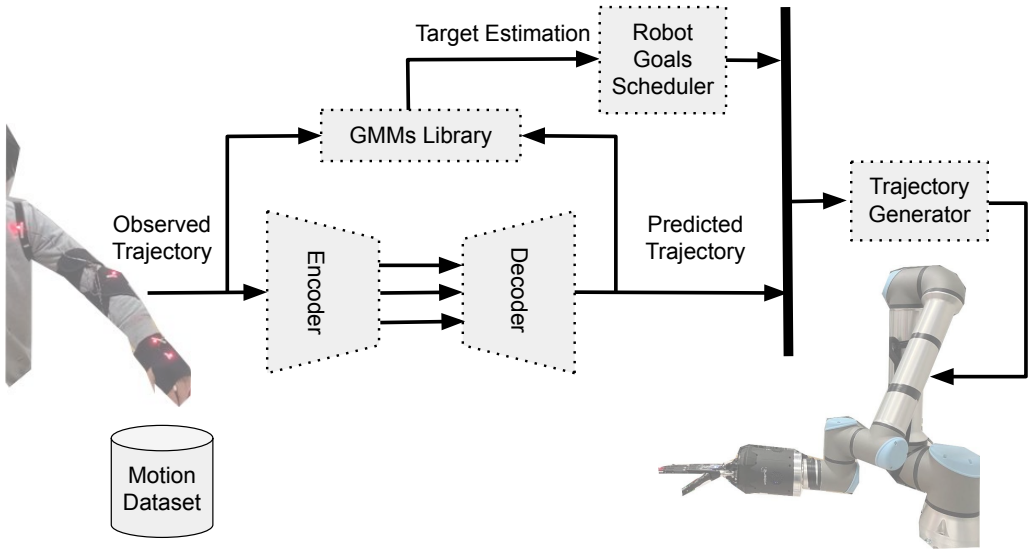

Fig. 4. The architecture of our proposed HRC pipeline. It contains three main parts: trajectory prediction, target estimation, and online trajectory generation. The encoder-decoder network takes the observed human arm trajectory as input and predicts the next steps of the human arm trajectory. The GMMs library then estimates the goal position of the human hand based on the observed and predicted trajectories. The encoder-decoder network and the GMMs library are both trained on a self-collected human arm motion dataset. The goal schedule module adjusts the robot goals based on human intentions. The trajectory generator yields collision-free trajectories during the collaboration tasks.

promoted in [17], where GMMs are used to approximate one class of trajectories. In [22], Landi et al. combined the minimum jerk model with an adaptive neural network to predict whether the human will react to the robot end-effector. The similarity between observed short-term movements and the learned user behavior was used to predict human reaching goal in a teleoperation task [23].

The Q-learning method was also used for this task. Cheng et al. [2] proposed that humans optimize a reward function during the pick-and-place task, related to the distance and velocity from the human hand to the target position. Assuming that the human motion follows a Boltzmann policy, they estimated the posterior probability distribution over all targets based on the observed trajectory. However, this method does not work so well when targets are located close to each other (e.g., 10 cm in our scenario), because there will be several similar probable target positions in this situation, especially during the initial motion stage.

Therefore, we will use the probabilistic model GMMs for target estimation instead of end-to-end deep learning methods. The benefit of GMMs is that they are easier to train and also provide us with probability information. As the target positions in our task are close to each other, the trajectories are very similar at the early stage. Unlike the work mentioned above, we make use of both the observed AH trajectory and the short-term prediction as the input of GMMs to improve the estimation accuracy at the beginning of the reaching motion.

C. Online Trajectory Generation

Only specific motion planning algorithms can deal with the dynamic obstacle avoidance problem, such as trajectory optimization [24] and sampling-based methods [25]. Considering the whole volume of obstacles across all prediction time steps for safe trajectory generation results in conservatively planned trajectories. Zheng et al. [11] propose a framework to deal with this problem. They reformulate the obstacle avoidance problem into two Quadratic Programming (QP) programs. This way, they can generate a collision-free trajectory very fast. However, in some scenarios, e.g., when the separating plane used in their approach is close to vertical, the generated trajectory is not safe anymore because of local minima and the linearized kinematics. In other work like [26], [27], they generated collision-free and custom-preferred waypoints in Cartesian space online, during which the dynamics limitation was not considered. They then control the robot end-effector to track these points.

In our recent work [28], we solved a trajectory tracking problem in a static environment. In this paper, we model a predicted AH trajectory as several moving capsules and solve the trajectory optimization problem in a model predictive control (MPC) style. A set of penalty terms are added into cost functions to efficiently generate a smooth and safe trajectory for the dynamic HRC task.

III. METHODOLOGY

As mentioned above, our proposed HRC pipeline is divided into three main parts, namely trajectory prediction, final target estimation, and online trajectory generation, as shown in Fig. 4.

A. Human Trajectory Prediction and Target Estimation

Predicting the human arm trajectory is the fundamental step in our system. The predicted human trajectory is not only employed to infer the intended hand position, but also enables the controller to generate a safe trajectory.

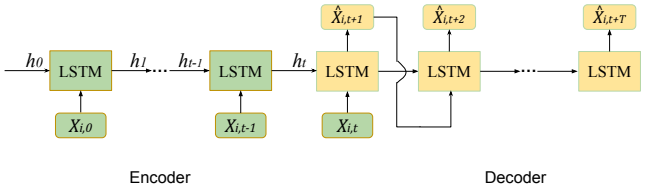

Fig. 5. The architecture of the encoder-decoder neural network. It consists of two modules, i.e, encoder and decoder. Both encoder and decoder modules are composed of LSTM cells.

To predict a trajectory, we use the encoder-decoder structure of the sequence-to-sequence (Seq2Seq) model similar to [6]. This encoder-decoder model is based on LSTM [29] which can account for dependencies in long sequence data. The architecture of the encoder-decoder network in this paper is shown as Fig. 5.

Although the trained GMMs described below could also be used for regression, the shape of the predicted trajectory is not similar to the actual trajectory [17]. Hence, we still make use of the Seq2Seq model for motion prediction.

The Seq2Seq model can be expressed as $\hat{X}_{i,t+1:t+T} = f(X_{i,0:t})$ and is trained on a dataset $\mathcal{D} = (X_i)_{i=1}^{N}$, where N is the number of demonstrated trajectories, t is the observed trajectory length, and T is the length of the predicted trajectory. $X_{i,t}$ are the positions of the human shoulder, elbow, wrist, and palm in Cartesian space. We split every trajectory into pieces, i.e., $X_{i,0:t+T}$. The input of the model is the observed trajectory $X_{i,0:t}$ and the labels are $X_{i,t+1:t+T}$. Our goal is to train this model to make the prediction $\hat{X}_{i,t+1:t+T}$ close to $X_{i,t+1:t+T}$. The loss function for the network is a weighted prediction error over four different markers, as follows:

$$Loss = \sum_{X_{i,0:T} \in D} \sum_{j=1}^{4} w_j ||\hat{X}_{i,t+1:T} - X_{i,t+1:T}|| \tag{1}$$

The weight values will be chosen by grid search. With the dataset $\mathcal{D}$, we also train a GMM library $G = (g_m)_{m=1}^{M}$ by the well-known unsupervised expectation maximization (EM) algorithm, where M represents the number of potential target objects. For the target position estimation task, we just use the trajectory of the human palm in the dataset. This is because the palm motion encodes the most related information for trajectory classification [17]. The observed trajectory X_{ob} is a $m \times n$ matrix where m is the number of waypoints and n is the dimensions per waypoint. We use K multivariate Gaussians $(gc_k)_{k=1}^{K}$ to approximate every g_m in G. So the probability of $X_{i,t}$, one trajectory point at time step t during demonstration i, belonging to g_m is given by:

$$p(X_{i,t}|g_m) = \sum_{k=1}^{K} p(gc_k|g_m) p(X_{i,t}|gc_k, g_m) \tag{2}$$

The probability of $X_{i,t}$ given gc_k and g_m is a Gaussian distribution:

$$p(X_{i,t}|gc_k, g_m) = \frac{1}{\sqrt{(2\pi)^n |\sigma_k|}} e^{-\frac{1}{2}(X_{i,t}-\mu_k)^T \Sigma_k^{-1} (X_{i,t}-\mu_k)} \tag{3}$$

Due to the similarity of the trajectories at the initial stage of the reaching motion, we will use both the observed and predicted trajectory for target position estimation. Then the probability of the observed trajectory $X_{i,0:t}$ together with the predicted trajectory $\hat{X}_{i,t+1:t+T}$ given g_m is seen in equation (4), where $X_{i,s} \in X_{i,0:t} \bigcup \hat{X}_{i,t+1:t+T}$.

$$p(X_{i,0:t}, \hat{X}_{i,t+1:t+T}|g_m) = \prod_{s=0}^{t+T} p(X_{i,s}|g_m) \tag{4}$$

According to the Bayesian rule, the log-likelihood of g_m given $X_{i,0:t}$ and $\hat{X}_{i,t+1:t+T}$ is given by equation (5). We will choose the g_m with the highest posterior probability as the intended target position.

$$p(g_m|X_{i,0:t}, \hat{X}_{i,t+1:t+T}) = \sum_{s=0}^{t+T} \log p(X_{i,s}|g_m) + \log p(g_m) \tag{5}$$

B. Online Trajectory Generation

Our optimization-based online trajectory generation method allows the robot to perform manipulation tasks while at the same time avoiding human AH motions, workspace boundaries, joint position limits, and dynamic constraints.

A limited quadratic position loss l_P is calculated from the distance between end-effector position P_E and goal position P_G. This loss is clipped at a maximum value m to avoid overriding other objectives such as collision avoidance. We also minimize a quadratic orientation loss l_R between end-effector orientation matrix R_E and goal orientation matrix R_G.

$$l_P = \min(m, \|P_E - P_G\|)^2 \tag{6}$$

$$l_R = \|R_E - R_G\|^2 \tag{7}$$

Robot joint angles $(p_i)_{i=0}^{T_R}$ are constrained by joint position limits $(p_{i,min}, p_{i,max})_{i=0}^{T_R}$, velocity limits $(v_{i,min}, v_{i,max})_{i=0}^{T_R}$ and acceleration limits $(a_{i,min}, a_{i,max})_{i=0}^{T_R}$, where T_R is the length of generated robot trajectory per optimization loop.

$$p_{i,min} < p_i < p_{i,max} \tag{8}$$

$$v_{i,min} < p_i' < v_{i,max} \tag{9}$$

$$a_{i,min} < p_i'' < a_{i,max} \tag{10}$$

During the real experiment, time delay exists due to calculation and communication between different modules. To prevent robot motion jumps because of the trajectory replacement between adjacent optimization loops, we constrain the first two steps of any newly generated trajectory to be the same as the corresponding steps in the previous trajectory.

We also add a velocity and acceleration regularizer r_i with weights b, c to prefer smooth motions.

$$r_i = b\, p_i'^{\,2} + c\, p_i''^{\,2} \tag{11}$$

Finally, we need to avoid collisions between the robot and humans as well as between the robot and the fixed workspace boundaries. We model human limbs and robot links as capsule-shaped collision objects, i.e., cylinders with hemispherical caps as shown in Fig. 6a. This results in a much

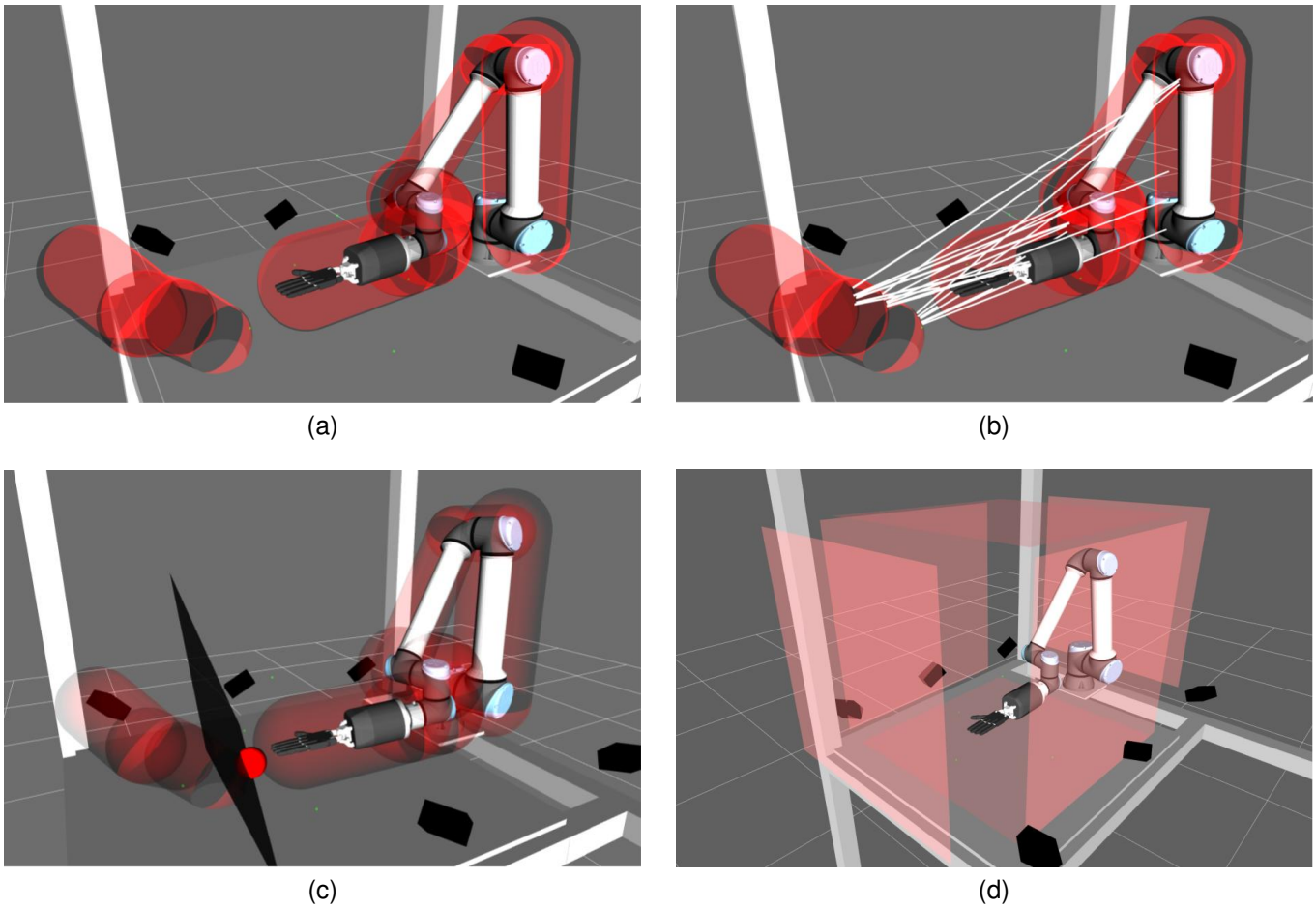

Fig. 6. Separation plane generation and workspace constraints for collision avoidance. **(a)** We use capsules as efficient collision geometries. Several capsules are created to cover the robot and the human arm. **(b)** The white lines visualize the pairwise closest distances between every robot capsule and human capsule. **(c)** Separation planes are calculated between every human capsule and robot capsule. These planes are used in our trajectory optimization to guarantee a safe separation whenever the distance between the human and robot is less than a threshold. **(d)** Static workspace boundaries are defined by six planes (four vertical planes and two horizontal planes) to restrict the robot motion to the defined volume.

lower number of collision geometries compared to the familiar representation using sets of collision spheres. Capsules with radius $(r_{H,j})_{j=0}^{T}$ are created between all connected human joints with predicted positions $(P_{H,j})_{j=0}^{T}$, and capsules with radius $(r_{R,k})_{k=0}^{T_R}$ are created between all connected robot joints with positions $(P_{R,k})_{k=0}^{T_R}$. Based on the method provided by [30], we compute the pairwise closest distances between the segments connecting robot joints and the segments connecting human joints. Then these closest distances subtract the radii of capsules are the pairwise closest distances between the human limb capsules and robot link capsules, as shown in Fig. 6b, and use the directions of the shortest distance vectors as separating plane normals $N_{j,k}$, as shown in Fig. 6c.

For human-robot collision avoidance, we add a set of penalty terms $q_{j,k}$ with the desired minimum distance d. We use penalty terms instead of hard constraints because the speed at which the human approaches the robot might be faster than the maximum velocity at which the robot is allowed to move. In such cases, with a hard constraint, the problem could become infeasible, and the robot might stop, provoking a collision. With a soft penalty, the robot will keep moving away from the human as quickly as possible. Since the position loss l_p is limited, collision avoidance still has precedence over reaching the target.

In many scenarios, a large open space will be available into which the robot can safely retreat. When avoiding human motions, it may be preferable for the robot to move towards this area. We therefore support an optional bias B. In our experiments, the robot can safely move upwards into a large open area above the table, so we set the bias B to $\left[0, 0, 0.5\right]$.

$$N_{B,j,k} = \frac{N_{j,k} + B}{\|N_{j,k} + B\|} \tag{12}$$

$$q_{j,k} = \min(0, N_{B,j,k}(P_{R,k} - P_{H,j}) - r_{H,j} - r_{R,k} - d)^2 \tag{13}$$

Finally, the workspace boundaries are enforced by a set of planes with normals $(N_{B,m})_{m=1}^{6}$ and offsets $(o_{B,m})_{m=1}^{6}$, as shown in Fig. 6d. We add one inequality constraint for each plane and robot sphere.

$$P_{R,k} N_{B,m} < o_{B,m} \tag{14}$$

At each time step, we optimize a trajectory with 10 future time steps and a step size of $0.1\,\mathrm{s}$. The trajectory is

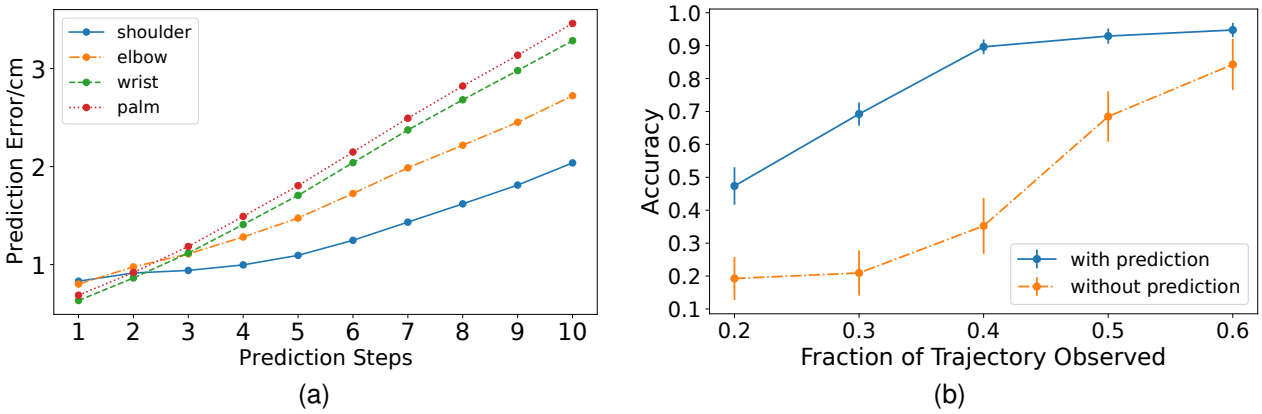

Fig. 7. Analysis of trajectory prediction and target estimation. **(a)** Mean trajectory prediction errors over different time horizons of the four tracked human arm and hand joints; **(b)** Target estimation accuracy at the early stage, with and without the short-term trajectory prediction. The result shows that the predicted trajectory is beneficial for early target estimation.

re-optimized at 10 Hz with the most recent human motion predictions. As the basis for our implementation, we solve the optimization problem via sequential quadratic programming using a primal-dual interior-point method described in [28].

IV. EXPERIMENTS

In this part, we evaluate the proposed method with real human motion data and the robot system in a desktop assembly task scenario as shown in Fig. 1. The results show that our method generates collision-free trajectories and helps the robot to collaborate with the human more efficiently and safely.

A. Trajectory Prediction and Target Position Estimation

Many devices have been developed for body tracking [31], [32]. In our experiment, four LED markers were placed on the human shoulder, elbow, wrist, and palm, respectively (Fig. 1). While the subject was performing the pick-and-place task, AH motions were recorded by a PhaseSpace Impulse X2 motion-capture system. We collected a dataset of pick-and-place trajectories from 5 healthy subjects (4 male, 1 female) of different body heights. The data were recorded at 270 Hz and afterwards re-sampled down to 27 Hz for the use of the Seq2Seq neural network. The number of recorded human trajectories per subject is 240, and 25% of the dataset per subject is held out for testing.

In the training phase of the Seq2Seq neural network, the dataset was split into equally sized trajectory pieces of 0.7s duration each. The former 0.35s part of the trajectory (10 steps) is the input of the neural network, and the remaining 0.35s trajectory (10 steps) is the label of the neural network output. One pick-and-place cycle consists of about 32 steps. Finally, the training dataset contains 48.6K samples, and the test dataset contains 14.6K samples.

See Fig. 2 for the layout of the twelve target positions. They are placed in two rows, six targets per row. Targets in the same row are at an interval of 10 cm, and the distance between these two rows is 20 cm. The ids for targets are numbered top-down and left-to-right from the human viewpoint, so 1..6 in the first and 7..12 for the second line.

During the initial training experiment, an LSTM layer with a 128-dimension hidden state was chosen. The Seq2Seq neural network is trained using Pytorch [33] with a batch size of 128 and a teacher forcing rate of 0.6. The learning rate is initialized to 0.005, with an exponential decay rate of 0.01. To speed up the training process, the batch normalization (BN) technique is also used. The weights in the loss function (hyperparameters) are 0.08, 0.16, 0.32, 0.44 by grid search.

The test results for trajectory prediction are plotted in Fig. 7a. We can see that the biggest prediction error is from the palm and wrist because the motions of these two joints vary a lot. The prediction errors for all joints are less than 2 cm over 5 time steps.

For predicting the human target position of the human reaching motion, twelve GMMs are trained with Python [34] on dataset $\mathcal{D} = (X_{i,0:T})_{i=1}^{N}$, with 75% used for training and 25% for testing. The hyperparameter K of GMMs was set to 18 based on the Bayesian Information Criterion(BIC) [35]. As mentioned above, the initial parts (about 40%) of the recorded human trajectories are very similar to each other, and the data points overlap, as shown in Fig. 3b. This makes it very difficult to classify the trajectory at the early stage, see Fig 7b. Based only on the observed motion waypoints, the GMM classification (orange curve) is initially mostly random, but improves after about 40% of the trajectories have been observed. Note that some false classifications remain even after 60% of the human motion is known.

However, our LSTM network has already learned to disambiguate between the different reaching motions and to predict the human palm positions for the next few time steps. Therefore, we can significantly improve the accuracy of the reaching target classification during an ongoing motion by feeding the GMMs with a few predicted hand positions in addition to the observed positions.

The resulting improvement of the classification as a function of the percentage of observed trajectory is shown in Fig. 7b (blue curve). To analyze the algorithm's performance in more detail, we also present the corresponding confusion matrices, see Fig. 8. The diagrams show the classification results without and with the LSTM predictions (upper / lower

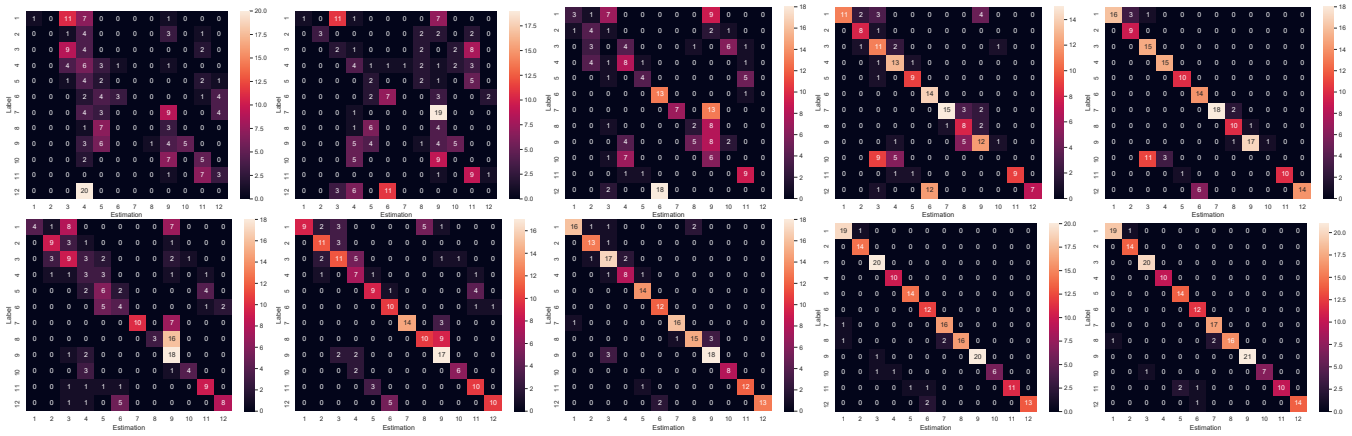

Fig. 8. The figures in the upper row show the confusion matrices for intended position estimation using only the observed trajectories. The figures in the lower row show the confusion matrices for intended position estimation using observed and predicted trajectories. Each column corresponds to the condition that 20%, 30%, 40%, 50%, and 60% of the reaching trajectory have been observed.

row) after $(20\%, 30\%, 40\%, 50\%, 60\%)$ of a human reaching trajectory is input to the GMMs. In the upper row, the initially rather random behavior can clearly be seen, only improving after 50% and 60% of the human motions are observed. Still, some false classifications between targets 3 and 10 and targets 6 and 12 remain; both are easily explained by the experiment layout, where the human hand passes over targets 3 and 6 to reach targets 10 and 12. With trajectory prediction enabled, much more accurate and robust estimates are obtained if at least 30% of the human motion has been observed.

B. Online Trajectory Generation

The proposed online trajectory generation method is tested on a 6-DoF manipulator UR10e arm with a Shadow C6 hand. The testing scenario is shown in Fig. 1. To ensure human safety, the maximum joint velocity of the arm is limited to $0.02\,\mathrm{rad\,s^{-1}}$ and acceleration are limited to $1\,\mathrm{rad\,s^{-2}}$ (velocity and acceleration constraints described above). The radius for every capsule is $10\,\mathrm{cm}$ and the collision margin d between capsules is set to zero.

The trajectory optimizer is implemented in C++ using the Eigen library for linear algebra functions. All of the modules communicate with each other via ROS [36] platform. We use MoveIt [37] to load the robot model, and Roscontrol [38] with the ur_modern_driver [39] to command the robot in real-time. We calculate the predicted LSTM and GMMs results at around $20\,\mathrm{Hz}$, while the trajectory optimizer runs at $10\,\mathrm{Hz}$.

This experiment has two phases: a reaching phase and a staying phase. In the reaching phase, the human takes one screw bolt from the initial position and puts it to target 3. Once the human moves from the initial position, the robot also starts to move from target 7 to the position of target 6. After the bolt is placed, the human stays at target 3, working for 5 seconds, while the robot is required to continue its task; this is the staying phase. Finally, the human and the robot return to their initial positions. To compare the performance between a reactive controller (considering only the current AH positions during robot trajectory generation) and the predictive controller (considering current and predicted AH positions during robot trajectory generation), we did the experiment multiple times, as shown in Fig. 9 and Fig. 10. In this scenario, the predictive controller uses five predicted palm positions to improve the target estimation at the early stage.

Fig. 9 shows the experimental results of the reactive controller. The figures from the first three columns show the human reaching phase. Due to the lack of human motion prediction, the robot starts moving towards its target, but also towards the human arm. When the distance between the human arm and robot becomes less than the predefined threshold (the third column of the figures), the robot automatically adjusts its motion to avoid the human arm and then continues to move to target 6. The overall motion is still collision-free of course, but far from optimal for the robot.

Fig. 10 presents the performance of predictive controller. As

TABLE I
COMPARISON RESULTS BETWEEN PREDICTIVE AND REACTIVE CONTROLLER

Trajectory Controller	Trial	Minimum Distance (cm)	Trajectory Length (cm)	Execution Time(s)
Predictive Controller (ours)	1	25.40	75.07	3.88
	2	26.58	86.73	3.89
	3	25.33	99.25	3.89
	Average	**25.77**	**87.02**	**3.89**
Reactive Controller (baseline)	1	18.31	138.72	4.64
	2	14.67	141.24	4.71
	3	16.88	144.15	4.95
	Average	16.62	141.37	4.77

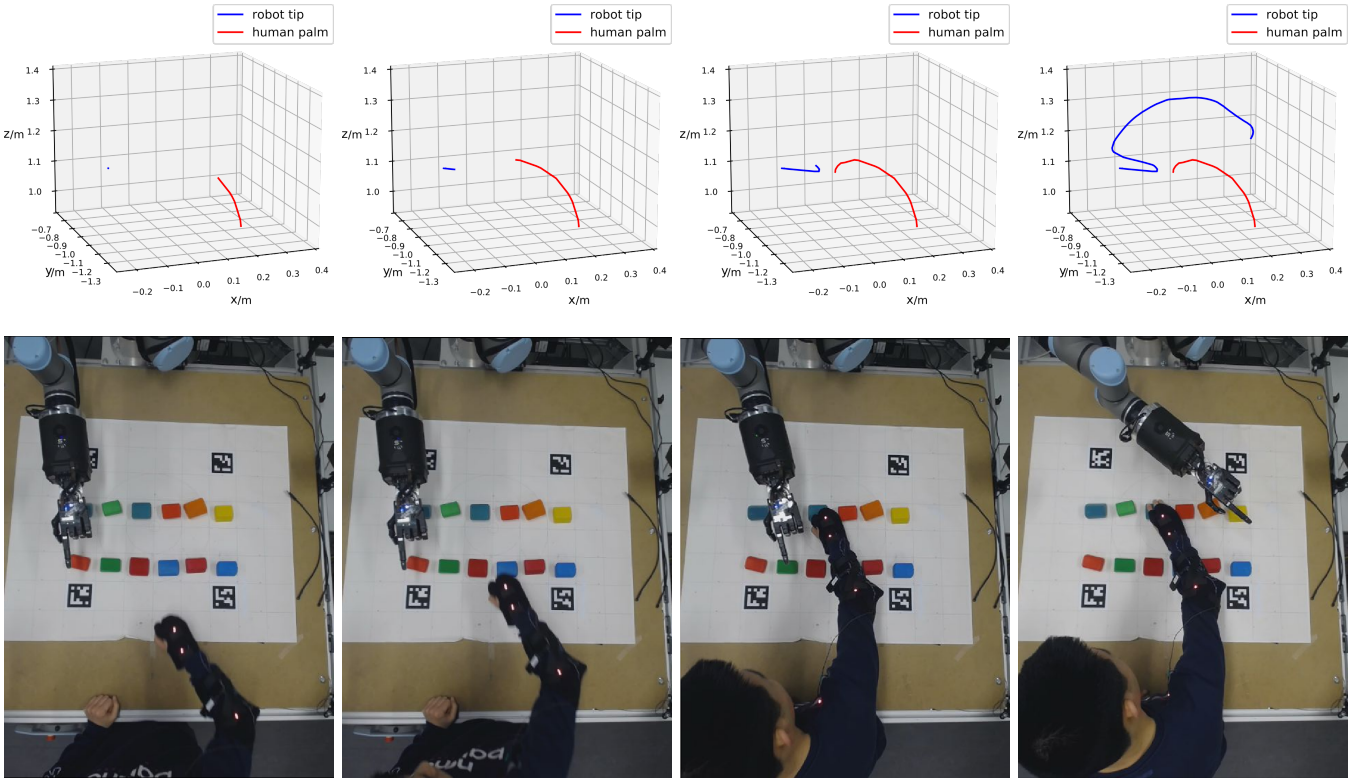

Fig. 9. Experiments with a reactive controller (no prediction). The figures in the upper row visualize Cartesian trajectories of the human and the robot. The blue line represents the trajectory of the robot (Shadow hand first finger tip, moving from target 7 to target 6), and the red line represents the trajectory of the human (palm joint, reaching target 3). The photos in the lower row show the human arm movement during the reaching phase, then the human arm stays at the target for 5s, as shown in the last figure.

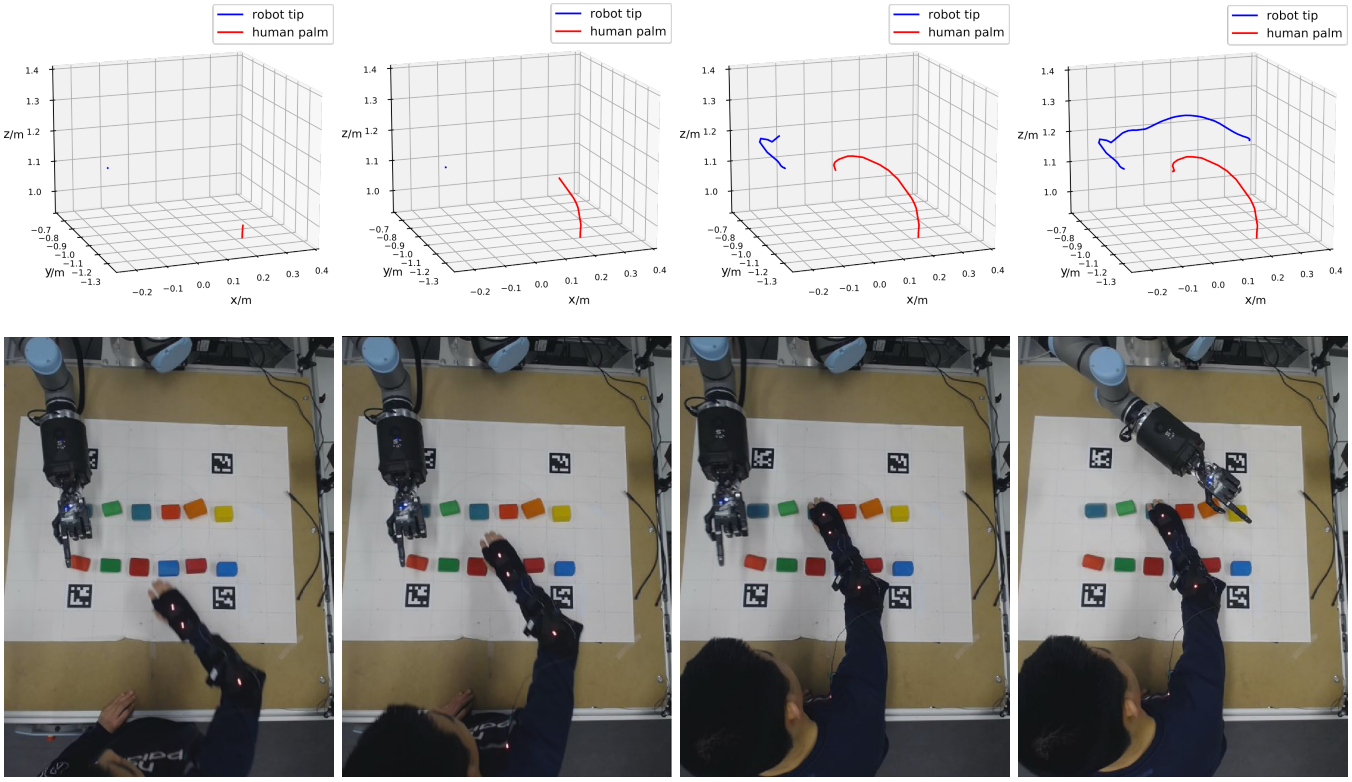

Fig. 10. Experiments with our predictive controller. Human motions, start and goal positions, as well dynamic constraints of the robot, are the same with Fig. 9

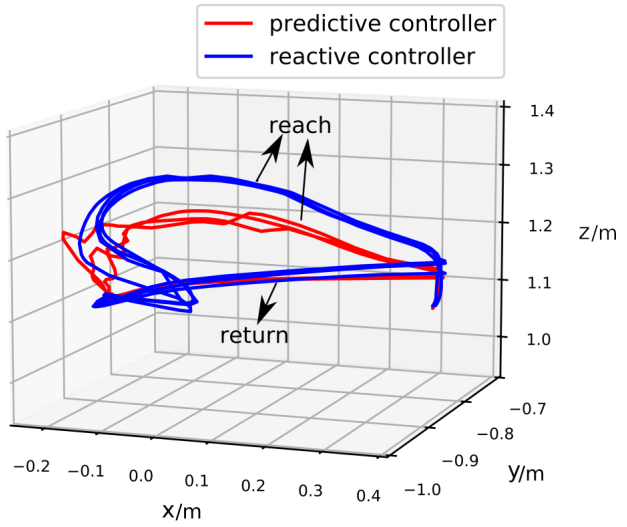

Fig. 11. The trajectories of robot Shadow hand first finger tip during the experiments with reactive and predictive trajectory controller.

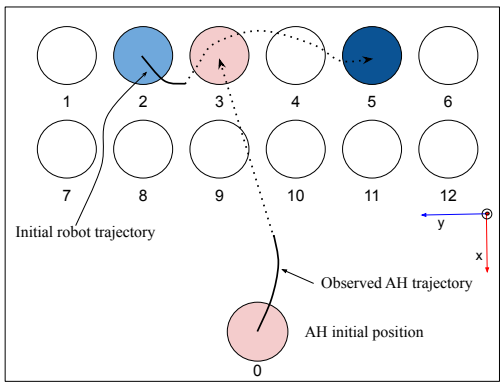

Fig. 12. Dynamic goal scheduling. Initially, the robot plans to go to target 3. Once the human intention and AH trajectory prediction based on the observed AH trajectory indicate target 3 for the human, the robot will replan its trajectory online (here to target 5 instead of target 3), keeping a safe distance so that it will not disturb the human task.

shown by the figures in the third column, the robot predicts much sooner that the human will enter the shared workspace, and the trajectory to target 6 is planned accordingly. Importantly, the closest distance between the human arm and the robot is also larger than that in Fig. 9, which means that the predictive controller is safer than the reactive controller.

For a direct comparison of the generated robot trajectories, Fig. 11 plots three successive task executions (reach, stay, return) for both controllers in a single diagram, where the large detour taken by the reactive controller is clearly visible for the reach phase. The trajectories from the predictive controller also include some noise initially, but become smooth once the target prediction remains stable. The return trajectories are almost optimal (straight lines) for both controllers.

We also analyzed the minimum distance between the human palm and the robot's index fingertip, the length of the robot trajectories, and the execution time under both controllers, see TABLE I. As expected, the trajectories generated by the

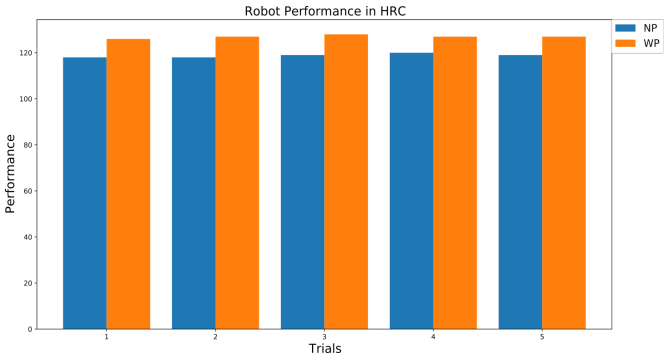

Fig. 13. The number of assembled products by the robot in two conditions. **(a)** NP: human intention classification with no predicted arm trajectory; **(b)** WP: human intention classification with predicted arm trajectory.

predictive controller are shorter, and the minimum distance between the human and robot with the predictive controller is larger than for the reactive controller. These results suggest that the predictive controller is more efficient and safer than the reactive controller.

C. HRC Efficiency

In order to investigate the efficiency of the whole algorithm, the participant was asked to carry out the collaboration task in more scenarios. During the experiments, the participant picked the object from the initial position and put the object in the 12 target positions in any random order, like 7→12→5→11→3→9→2→8→10→4→1→6→7 in this experiment. One example to explain the task flow is that the person placed a screw bolt in a target position, like target 3, and waited there for 5s. Meanwhile, the robot needed to change its target position from target 3 to target 5 and replan its trajectory based on the predicted AH trajectory as shown in Fig. 12, to avoid disturbing the human's work. The relationships between the human intended target position and robot available target positions are shown as TABLE II. The main idea of the workflow design is to let the robot and the human work concurrently without interfering with each other during the collaboration process. The workflow is coordinated by the robot goal scheduler module in Fig. 4. Once the robot finished touching target 6, it continued to touch

TABLE II
EXPERIMENT WORKFLOW

Human Intended Target	Robot available Targets
1	3, 4, 5, 6, 10, 11, 12
2	4, 5, 6, 7, 11, 12
3	1, 5, 6, 7, 12
4	1, 2, 6, 7, 8, 12
5	1, 2, 3, 7, 8, 9
6	1, 2, 3, 4, 7, 8, 9, 10
7	1, 2, 3, 4, 5, 6, 9, 10, 11, 12
8	1, 2, 3, 4, 5, 6, 10, 11, 12
9	1, 2, 3, 4, 5, 6, 7, 11, 12
10	1, 2, 3, 4, 5, 6, 7, 8, 12
11	1, 2, 3, 4, 5, 6, 7, 8, 9
12	1, 2, 3, 4, 5, 6, 7, 8, 9, 10

targets in the order 7→8→9→10→11→12→1 if there were no human movements in the workspace. Otherwise, the robot target position was scheduled online according to TABLE II. We did this experiment in 2 different situations for the ablation study.

In situation 1, we predicted human intention (human target position) without any AH motion prediction (NP) and replanned the robot trajectory with the predictive controller. In situation 2, human intentions were estimated with a predicted AH trajectory (WP). The trajectory of the robot was also replanned with the predictive controller. During the total duration of the experiment, we counted the number of assembled products by the robot. From Fig. 13, we can see that the HRC efficiency was reliably improved based on our algorithm strategy. The improvement was not significant, and one reason was that many human-robot target combinations are conflict-free (e.g., human's intended target was at 7, and the robot was going from target 3 to target 4.). If the conflict-free human-robot target combinations were excluded, the HRC efficiency improvement between WP and NP would be more significant.

V. CONCLUSION AND FUTURE WORK

We proposed a pipeline to improve the efficiency and safety of HRC assembly tasks. We trained a Seq2Seq neural network to predict the human AH trajectory accurately. Unlike other methods, we made use of both the observed and predicted trajectory as the input of GMMs for target estimation. As shown by our experiments, this results in a much more accurate posterior probability distribution over all potential target positions from the early stages of human motion, even if the trajectories are initially very similar to each other.

The predicted trajectory and estimated target position were then combined to generate a goal-oriented and collision-free trajectory based on a novel trajectory generation method. We evaluated the effectiveness of the whole pipeline on our real robot system, and the results demonstrate an enhanced safety and efficiency of the HRC task.

For future work, more experiments need to be conducted to test the pipeline's effectiveness in more complex scenarios. We also plan to replace the motion-capture system and use the raw data from cheap RGB-D cameras for human trajectory prediction and intention recognition. It will also be interesting to fuse more information, such as human gaze or semantic task information, to improve the target estimation accuracy and robustness.

REFERENCES

[1] L. Fu and J. Zhao, "Maxwell-Model-Based Compliance Control for Human–Robot Friendly Interaction," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 13, no. 1, pp. 118–131, 2020.

[2] Y. Cheng, L. Sun, C. Liu, and M. Tomizuka, "Towards efficient human-robot collaboration with robust plan recognition and trajectory prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2602–2609, 2020.

[3] Y. Wang, Y. Sheng, J. Wang, and W. Zhang, "Optimal collision-free robot trajectory generation based on time series prediction of human motion," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 226–233, 2017.

[4] W. Zhao, L. Sun, C. Liu, and M. Tomizuka, "Experimental evaluation of human motion prediction toward safe and efficient human robot collaboration," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 4349–4354.

[5] 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.

[6] H. Wang, J. Dong, B. Cheng, and J. Feng, "PVRED: A Position-Velocity Recurrent Encoder-Decoder for Human Motion Prediction," *IEEE Transactions on Image Processing*, vol. 30, pp. 6096–6106, 2021.

[7] L.-Y. Gui, Y.-X. Wang, D. Ramanan, and J. M. Moura, "Few-shot human motion prediction via meta-learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 432–450.

[8] R. C. Luo and L. Mai, "Human intention inference and on-line human hand motion prediction for human-robot collaboration," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5958–5964.

[9] J. S. Park, C. Park, and D. Manocha, "I-planner: Intention-aware motion planning using learning-based human motion prediction," *The International Journal of Robotics Research*, vol. 38, no. 1, pp. 23–39, 2019.

[10] S. Brossette and P.-B. Wieber, "Collision avoidance based on separating planes for feet trajectory generation," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 509–514.

[11] P. Zheng, P.-B. Wieber, and O. Aycard, "Online optimal motion generation with guaranteed safety in shared workspace," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9210–9215.

[12] S. Qiu, W. Guo, D. Caldwell, and F. Chen, "Exoskeleton online learning and estimation of human walking intention based on dynamical movement primitives," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 13, no. 1, pp. 67–79, 2020.

[13] J. Mainprice, R. Hayne, and D. Berenson, "Goal set inverse optimal control and iterative replanning for predicting human reaching motions in shared workspaces," *IEEE Transactions on Robotics*, vol. 32, no. 4, pp. 897–908, 2016.

[14] A. Møgelmose, M. M. Trivedi, and T. B. Moeslund, "Trajectory analysis and prediction for improved pedestrian safety: Integrated framework and evaluations," in *2015 IEEE intelligent vehicles symposium (IV)*. IEEE, 2015, pp. 330–335.

[15] F. Bartoli, G. Lisanti, L. Ballan, and A. Del Bimbo, "Context-aware trajectory prediction," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 1941–1946.

[16] F. Kuhnt, R. Kohlhaas, T. Schamm, and J. M. Zöllner, "Towards a unified traffic situation estimation model—street-dependent behaviour and motion models," in *2015 18th International Conference on Information Fusion (Fusion)*. IEEE, 2015, pp. 1223–1229.

[17] R. Luo, R. Hayne, and D. Berenson, "Unsupervised early prediction of human reaching for human–robot collaboration in shared workspaces," *Autonomous Robots*, vol. 42, no. 3, pp. 631–648, 2018.

[18] A. Maye, D. Zhang, and A. K. Engel, "Utilizing retinotopic mapping for a multi-target SSVEP BCI with a single flicker frequency," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 7, pp. 1026–1036, 2017.

[19] S. Trick, D. Koert, J. Peters, and C. A. Rothkopf, "Multimodal uncertainty reduction for intention recognition in human-robot interaction," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 7009–7016.

[20] B. Fang, W. Ding, F. Sun, J. Shan, X. Wang, C. Wang, and X. Zhang, "Brain-computer interface integrated with augmented reality for human-robot interaction," *IEEE Transactions on Cognitive and Developmental Systems*, 2022.

[21] 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.

[22] C. T. Landi, Y. Cheng, F. Ferraguti, M. Bonfè, C. Secchi, and M. Tomizuka, "Prediction of human arm target for robot reaching movements," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5950–5957.

[23] L. Wang, Q. Li, J. Lam, Z. Wang, and Z. Zhang, "Intent inference in shared-control teleoperation system in consideration of user behavior," *Complex & Intelligent Systems*, pp. 1–11, 2021.

[24] J. Dong, M. Mukadam, F. Dellaert, and B. Boots, "Motion Planning as Probabilistic Inference using Gaussian Processes and Factor Graphs." in *Robotics: Science and Systems*, vol. 12, 2016, p. 4.

[25] K. Hauser, "On responsiveness, safety, and completeness in real-time motion planning," *Autonomous Robots*, vol. 32, no. 1, pp. 35–48, 2012.
[26] O. Brock and O. Khatib, "Elastic strips: A framework for motion generation in human environments," *The International Journal of Robotics Research*, vol. 21, no. 12, pp. 1031–1052, 2002.
[27] N. Sayols, A. Sozzi, N. Piccinelli, A. Hernansanz, A. Casals, M. Bonfè, and R. Muradore, "Global/local motion planning based on dynamic trajectory reconfiguration and dynamical systems for autonomous surgical robots," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8483–8489.
[28] P. Ruppel and J. Zhang, "Learning Object Manipulation with Dexterous Hand-Arm Systems from Human Demonstration," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5417–5424.
[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[30] D. Sunday, "Practical geometry algorithms: With c++ code," *KDP Print US*, 2021.
[31] B. Fang, X. Wei, F. Sun, H. Huang, Y. Yu, and H. Liu, "Skill learning for human-robot interaction using wearable device," *Tsinghua Science and Technology*, vol. 24, no. 6, pp. 654–662, 2019.
[32] B. Fang, F. Sun, H. Liu, C. Liu, and D. Guo, *Wearable technology for robotic manipulation and learning*. Springer, 2020.
[33] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, pp. 8026–8037, 2019.
[34] G. Van Rossum, F. L. Drake *et al.*, *Python reference manual*. iUniverse Indiana, 2000.
[35] G. Schwarz, "Estimating the dimension of a model," *The annals of statistics*, pp. 461–464, 1978.
[36] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng *et al.*, "ROS: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.
[37] D. Coleman, I. Sucan, S. Chitta, and N. Correll, "Reducing the barrier to entry of complex robotic software: a Moveit! case study," *arXiv preprint arXiv:1404.3785*, 2014.
[38] S. Chitta, E. Marder-Eppstein, W. Meeussen, V. Pradeep, A. R. Tsouroukdissian, J. Bohren, D. Coleman, B. Magyar, G. Raiola, M. Lüdtke *et al.*, "ros_control: A generic and simple control framework for ROS," *The Journal of Open Source Software*, vol. 2, no. 20, pp. 456–456, 2017.
[39] T. T. Andersen, "Optimizing the universal robots ros driver." Technical University of Denmark, Department of Electrical Engineering, Tech. Rep., 2015.

Jianzhi Lyu received his B.S. degree with honors in marine engineering from Huazhong University of Science and Technology, Wuhan, China, in 2014, and his M.S. degree in control engineering from Northwestern Polytechnical University, Xi'an, China, in 2017.

He is currently pursuing his Ph.D. degree in computer science with TAMS Group, Department of Informatics, Universität Hamburg, Germany. His research interests include model-based and data-driven methods for human-robot collaboration, dexterous manipulation in dynamic environment.

Philipp Ruppel received a M.Sc. degree in Informatics from Universität Hamburg, Germany, in 2017. He currently works as a researcher at Universität Hamburg and pursues a doctoral degree. His research interests include dexterous manipulation, optimization-based control, tactile sensing, and machine learning.

Norman Hendrich received the B.Sc., M.Sc. degrees in physics, and the Ph.D. degree in computer science from the Universität Hamburg, Hamburg, Germany, in 1986, 1991, and 1996, respectively.

He is currently a Senior Lecturer with the Department of Informatics, Universität Hamburg. He participated as a Principal Investigator in several collaborative European research projects and currently acts as a Project Manager of the joint Sino-German project Transregio-SFB TRR169 "Crossmodal Learning." His research interests include computer simulation and machine learning, with a focus on applications in service robotics and dexterous manipulation.

Shuang Li is currently pursuing the Ph.D. degree in computer science at Universität Hamburg. She received the bachelor's degree and master's degree in mechanical engineering from Anhui University of Technology. Her research interests are robotic teleoperation, dexterous manipulation and deep learning applications in robotics.

Michael Görner received a B.Sc. in Informatics from University of Potsdam, Germany, in 2012 and a M.Sc. degree in Cognitive Science with distinction from Osnabrück University, Germany, in 2016. He currently pursues a Ph.D. degree in computer science at Universität Hamburg. Their research interests include autonomous manipulation and manipulation planning in semi-structured environments for physical robot systems.

Jianwei Zhang (Member, IEEE) is professor and director of TAMS, Department of Informatics, Universität Hamburg, Germany. He received both his Bachelor of Engineering (1986, with distinction) and Master of Engineering (1989) at the Department of Computer Science of Tsinghua University, Beijing, China, his Ph.D. (1994) at the Institute of Real-Time Computer Systems and Robotics, Department of Computer Science, University of Karlsruhe, Germany.

His research interests are sensor fusion, intelligent robotics and multimodal machine learning, etc. In these areas he has published about 400 journal and conference papers, technical reports and four books. He is the coordinator of the DFG/NSFC Transregional Collaborative Research Centre SFB/TRR169 "Crossmodal Learning" and several EU robotics projects. He has received multiple best paper awards. He is the General Chair of IEEE MFI 2012, IEEE/RSJ IROS 2015, and the International Symposium of Human-Centered Robotics and Systems 2018. Jianwei Zhang is life-long Academician of the Academy of Sciences in Hamburg Germany.