2024 IEEE International Conference on
Advanced Robotics and Its Social Impacts (ARSO)
Hong Kong, China, May 20-22, 2024

2024 IEEE International Conference on Advanced Robotics and Its Social Impacts (ARSO) | 979-8-3503-4463-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ARSO60199.2024.10557849

Task Constraint-Guided Inverse Reinforcement Learning (TC-IRL) in Human-Robot Collaborative Assembly

Yi Chen, Weitian Wang and Yunyi Jia

Abstract — **Collaborative robots transit from the traditional robot-in-a-cell scenarios to a human-robot-shared workspace. This demands robots to better understand their human partners and then assist them. Existing robot learning from demonstration work mainly focuses on enabling robots to repeat human demonstrated tasks alone and usually require significant training efforts but have limited scalability to new tasks. This paper proposes a new task constraint-guided inverse reinforcement learning (TC-IRL) approach to learn assembly tasks from human demonstrations with significantly reduced state and action space (leading to less training data requirement) and computational efforts (landing to better real-time performance) than the conventional IRL. The TC-IRL is also extended to new geometric-scaled tasks to generate robot assistance to human in collaborative assembly. The proposed approaches are validated and evaluated through human-robot collaborative assembly experiments.**

I. Introduction

Collaborative robots are designed to work safely with humans in a shared workspace, which requires robots to accomplish flexible tasks in a changeable environment. To reduce the robot programming effort and improve the efficiency, the approaches of robot learning from demonstrations (RLfDs) [1] are investigated as feasible solutions for flexible tasks. Various techiques of demostration have been studied, such as kinesthetic demonstration [2], motion-sensor demonstration [3], and teleoperateed demonstration [4]. Abu-Dakka et al. [5] demonstrated peg-in-hole tasks on KUKA LWR arm in gravity comenpasation mode via kinesthetic guiding. In the demonstration, the human repeately guided the robot's tool center point along the disired trajectory that can successfully inserts the peg into the hole, meanwhile, the Cartesian space trajectories and the joint torques are recorded. Edmonds et al. [6] used a glove with taqctile sensors and IMUs to record both forces and positions of human hand in demonstrations of opening medicine bottle. Ferreira et al. [7] developed a vision-based real-time tool tracking system to record the tool trjectories in six degree-of-freedoms in human demonstrations, which can be used to program the industrial robots.

After task information in human demonstrations is captured, different learning models have been proposed to interpret the tasks for robot learning. For instance, the hidden Markov model (HMM) was applied to encode the demonstrated trajectories [8] as well as force-based behaviors [9]. Goto et al. [10] applied the finite state machine (FSM) to formulate a human-robot collaborative assembly task. Wang et al. [11] developed the assembly graph (AG) to interpret the pre-defined constraints of assembly tasks. Kim et al. [12] implemented the Gaussian Mixture Model (GMM) to model robot motions. However, these approaches mainly focus on making robots repeat the tasks that humans have demonstrated and lack of scalability to new tasks.

Recently, some studies have applied inverse reinforcement learning (IRL) on task learning from human demonstrations, which use reward functions to capture human working patterns [13]–[16]. However, due to the large state and action space of IRL, it usually requires a large amount of training data and computational efforts.

To address the above challenges of existing approaches, this paper proposes a new task constraint-guided inverse reinforcement learning (TC-IRL) approach to make collaborative robots learn the tasks from human teaching demonstrations and then assist humans to collaboratively accomplish the tasks including new tasks with larger geometric scales instead of repeating the learned tasks. Compared to conventional IRL, it can significantly reduce the state and action space and computational efforts, and therefore lead to less training data requirement and better real-time performance. Furthermore, a robot assistance generation approach with task extension is then proposed to generate assistive robot actions to collaborate with humans to accomplish not only the demonstrated tasks but also new tasks with larger geometric scales.

The proposed approaches potentially allow humans to teach the robot by just a few small-scale demonstrations and then the robot can assist humans to accomplish a series of larger-scale tasks in the human-robot collaboration process.

II. Human-Robot Collaborative Assembly

In the conventional inverse reinforcement learning (IRL) approach [13], an assembly task can be modeled by a Markov decision process (MDP), which can be described by a tuple as

$$M = (S, A, T, \gamma, R) \tag{1}$$

where S represents the assembly state space, A represents the assembly action space, $T = P(s' \mid s, a)$ is the state transition probability, $\gamma \in [0,1)$ is the discount factor, and R is the reward function.

In this IRL formulation, task constraints are connotative in the definitions of task states. Generally, all the potential states

Yi Chen is with ABB Robotics, USA (e-mail: yi.chen1@us.abb.com). (e-mail: yc4@clemson.edu), Weitian Wang is with the Department of Computer Science at Montclair State University, USA (e-mail: wangw@montclair.edu), and Yunyi Jia is with the Department of Automotive Engineering at Clemson University, USA (e-mail: yunyij@clemson.edu).

and actions that satisfy the task constraints must be defined in the model. For collaborative assembly applications, the actions and states in the MDP depend on parts, tools, assembly locations, sequences, etc. Therefore, the size of the action and state space will increase dramatically when the options of parts, tools, and task scales are slightly increased. As known, the size of the state and action space is proportional to the number of unknown parameters that need to be learned through IRL, and the number of unknown parameters indicates the required amount of data for training of IRL. In addition, the size of the state and action space is also proportional to the computational costs of implementing IRL. Therefore, when applying conventional IRL to learning assembly tasks, due to the wide variety of the parts, tools, and assembly task variations, it will require a very large state and action space and therefore require a large amount of data for training and significant computational efforts for implementation. This makes it difficult to be applied to realistic assembly scenarios. To address this issue, we propose the task constraint-guided inverse reinforcement learning (TC-IRL) approach. To formulate the human-robot collaborative assembly task, the set of tools can be defined as

$$Ts = \{ts_1, ts_2, ..., ts_{N_{ts}}\} \tag{2}$$

where N_{ts} is the total number of tools.

The set of parts to be assembled is defined as

$$Pt = \{pt_1, pt_2, ..., pt_{N_{pt}}\} \tag{3}$$

where N_{pt} is the total number of parts. The parts are distinguished from each other by a set of attributes, such as shape, color, mass, etc.

The set of the attributes that are used to describe different parts is defined as

$$Ar = \{ar_1, ar_2, ..., ar_{N_{ar}}\} \tag{4}$$

where N_{ar} is the total number of attributes. Each attribute can have different values, which can be written as

$$V_{ar_i} = \{{}^{ar_i}v_1, {}^{ar_i}v_2, ..., {}^{ar_i}v_{N_{ar_i}}\} \tag{5}$$

where $ar_i \in Ar$ is an attribute, N_{ar_i} is the total number of values corresponding to the attribute ar_i.

The set of assembly locations can be written as

$$Lc = \{lc_1, lc_2, ..., lc_{N_{lc}}\} \tag{6}$$

where N_{lc} is the total number of assembly locations. The number of locations in the human-robot collaboration phase can be variant and different from that in the human teaching phase. The former location set depends on how the human worker wants to extend the task scale, while the later location set is generally the minimal scale of human demonstrations, which are enough to teach all the task constraints and human preferences. In this paper, we present the derivation of the proposed approaches in two-dimensional assembly scenarios.

III. Learning Collaborative Assembly Tasks via TC-IRL

In this section, we present the detail of the task-constraint-guided inverse reinforcement learning (TC-IRL). A general form of the MDP in the TC-IRL can be written as

$$M_t = (S_t, A_t, T_t, \gamma, R_t) \tag{7}$$

where S_t is the task constraint-guided state space, A_t is the task constraint-guided action space, $T_t = P(s_t' | s_t, a_t)$ is the state transition probability, $\gamma \in [0,1)$ is the discount factor, and R_t is the task constraint-guided reward function.

In TC-IRL, the robot first learns the task constraints from human demonstrations, and the learned task constraints are then used to construct task constraint-guided state and action space which is much smaller than the original space. The task constraint-guided reward is then defined based on this constrained space and learning is conducted to learn the unknown parameters for assembly tasks from human demonstrations. In the following, we will first introduce the learning of task constraints including object-based task constraints, location-based task constraints, and human hand-based task constraints which are used to limit the size of the state and action space and then introduce the learning algorithm to learn how the human accomplishes the task.

A. Learning of Task Constraints

1) Object-based Task Constraints

The object-based constraints are defined to represent the part-tool pairs, which indicate the corresponding tools for each part. Based on the definitions in the previous section, the object-based constraints can be represented by a matrix:

$$C_{obj} = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,N_{pt}} \\ \vdots & \vdots & \cdots & \vdots \\ c_{N_{ts},1} & c_{N_{ts},2} & \cdots & c_{N_{ts},N_{pt}} \end{bmatrix}_{N_{ts} \times N_{pt}} \tag{8}$$

where the i^{th} row of C_{obj} corresponds to the tool ts_i in the set Ts, while the j^{th} column of C_{obj} corresponds to the part pt_j in the set Pt. To indicate the object-based constraints, the element $c_{i,j} = 1$ if the part pt_j should be assembled with the tool ts_i, otherwise, $c_{i,j} = 0$.

The object-based constraints can be learned by tracking the human hand motions and the state of parts/tools in the human demonstrations. The combined state at any given time can be represented as

$$s_h(t) = [obj_L(t), obj_R(t)]^T \tag{9}$$

where $obj_L(t)$ and $obj_R(t)$ indicate the object in the human's left and right hand at the moment $T = t$. To simplify the problem, we assume that each hand can be with either a part or a tool in hand or remain empty at each moment. Based on human operations in the demonstrations, a statistic-based approach is used to learn the object-based constraints.

For the k^{th} demonstration, let ${}^{d}L_k$ be the overall length of the state sequence in the demonstration, A_{1k} be the times of appearances of $s_h=[ts_i, pt_j]^T$, A_{2k} be the times of appearances of $s_h=[pt_j, ts_i]^T$, and N_D be the total number of demonstrations. The probability that the part pt_j should be assembled with tool ts_i can be written as

$$P(ts_i, pt_j)=\frac{1}{N_D}\sum_{k=1}^{N_d}\frac{A_{1k}+A_{2k}}{L_i} \tag{10}$$

This probability is then used to update each element in C_{obj} where for each part pt_j in the set Pt, the tool with the highest probability is considered the object-based constraint, and the corresponding element is set as 1.

2) Location-based Task Constraints

The location-based constraints are used to describe the requirements that any given assembly location must have the corresponding part installed in the final assembly. The location-based constraint of a two-dimensional assembly scenario can be written in matrix format as

$$C_{loc}=\begin{bmatrix} o_{1,1} & o_{1,2} & \cdots & o_{1,W} \\ \vdots & \vdots & \cdots & \vdots \\ o_{H,1} & o_{H,2} & \cdots & o_{H,W} \end{bmatrix}_{H\times W} \tag{11}$$

where H and W are the height and width for the demonstrated tasks and $o_{i,j}$ is a vector, which indicates the values of attributes of the object assembled at the corresponding assembly location.

Considering all the parts that have been installed at a given assembly location throughout multiple human demonstrations. The probability distribution for a specific attribute ar_i at the assembly location (x,y) can be calculated by

$$P(ar_i=v_j \mid X=x, Y=y)=\frac{count(ar_i=v_j)}{N_{\mathrm{D}}} \tag{12}$$

where $count(\bullet)$ function means to count the times of appearance of the given condition. N_D means the total number of rounds of human demonstrations.

Let p_{he} be the probability of human demonstration errors. For the part at assembly location (x,y), the attribute ar_i should have the value v_j as the constraint if it satisfies

$$P(ar_i=v_j \mid X=x, Y=y)>1-p_{he} \tag{13}$$

This probability is then used to update the element in C_{loc} where the attributes whose probabilities satisfy (12) are all considered as constraints in $o_{i,j}$.

3) Human Hand-based Task Constraint

Different human workers may have different hand preference for specific parts or tools. In order to deliver parts and corresponding tools to proper human hands based on their preference, the human hand-based task constraint with respect to part-tool pairs is defined, which can be represented by a matrix

$$C_{hand}=\begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,N_{pt}} \\ \vdots & \vdots & \cdots & \vdots \\ h_{N_{ts},1} & h_{N_{ts},2} & \cdots & h_{N_{ts},N_{pt}} \end{bmatrix}_{N_{ts}\times N_{pt}} \tag{14}$$

The i^{th} row corresponds to the tool ts_i in the set Ts, while the j^{th} column corresponds to the part pt_j in the set Pt. For those part-tool pairs, which are not satisfied with the object-based constraints or have never been appeared in any of human demonstrations, the corresponding element is set as $h_{i,j}=0$. The element $h_{i,j}=1$ if the part pt_j should be delivered to the right hand, while the tool should be delivered to the left hand. The element $h_{i,j}=-1$ represents the opposite hand preference

With the definitions of variables in (10), for each specific part-tool pair (ts_i, pt_j), the probability of different hand over methods among all the human demonstrations can be computed by

$$\begin{aligned} P(L=ts_i, R=pt_j)&=\frac{1}{N_D}\sum_{k=1}^{N_d}\frac{A_{1k}}{L_i} \\ P(L=pt_j, R=ts_i)&=\frac{1}{N_D}\sum_{k=1}^{N_d}\frac{A_{2k}}{L_i} \end{aligned} \tag{15}$$

The higher probability in (15) is regarded as the human hand preference on this specific part-tool pair (ts_i, pt_j). The values, -1 or 1, are signed to the corresponding elements in C_{hand} based on the probability.

B. IRL Learning of Assembly Tasks with Tasks Constraints

In order to learn how human accomplishes the assembly task, the state and action space in (7) is firstly constructed. Both the learned object-based and location-based task constraints will help limit the size of the state space S_t because only the constrained part-tool pairs $c_{i,j}$ learned in C_{obj} and only parts whose attributes satisfy the constraint vector $o_{i,j}$ in C_{loc} are considered in the state definition. The human hand-based task constraint will help limit the action space A_t because for a specific part or tool, which hand to use in the action will be specified in the constraint C_{hand}. With the definition of constrained state space and action space, the size of the entire state and action space (i.e., different action options at different states) will then be significantly reduced.

Additionally, the learning of assembly tasks is not only to learn the final assembly state but also to learn the process of how the human conduct the assembly during demonstrations. This process can be captured by a set of feature functions defined by

$$f=[f_1, f_2, ..., f_k, ...]^T \tag{16}$$

where each f_k is defined in the state and action space $f_k(s) \in \{0,1\}$ to specify a special feature of the human assembly process. Each different value of the vector f is corresponding to a human working style in the assembly, such as assembly the part row by row, left to right, from far to near with respect to his/her body position, etc.

The overall human assembly process can be then reflected by a task constraint-guided reward function which is defined as a weighted sum of the feature functions

$$R_t(s) = W^T f(s,a) = \sum_k w_k f_k(s,a) \tag{17}$$

where $W = [w_1, w_2, ...]^T$ is a set of weights to determine the preferences of the human on different features during assembly. The weights together will determine how the human would like to accomplish the assembly tasks.

In this paper, we propose to apply the maximum entropy inverse reinforcement learning (MaxEnt-IRL) [18] to learn the weights in the reward function from human demonstrations. We assume that the MDP is deterministic in this work. Therefore, according to the MaxEnt-IRL principle, the distribution over assembly strategy under deterministic transitions can be defined as

$$P(\zeta \mid M_t, W) = \frac{1}{Z(W)} \exp\left(\sum_k w_k{}^T f_k(s,a) \right) \tag{18}$$

where ζ is the assembly strategy in human demonstrations, $Z(W)$ is the partition function. The weights of features in the reward function can then be optimized by maximizing the entropy through

$$\begin{aligned} W^* &= \arg\max_W \log P(s \mid M, W) \\ &= \arg\max_W \left(\sum_k w_k{}^T f_k(s) - \log Z(W) \right) \end{aligned} \tag{19}$$

IV. TC-IRL Guided Robot Assistance

In this section, we will introduce how to generate assistive robot actions to collaborate with humans to accomplish not only the demonstrated tasks but also new tasks with larger geometric scales. The robot will ask the human through natural language about the dimensions of the new scaled tasks and the human will respond through natural language before the collaboration starts.

A. Extension of TC-IRL

When the collaborative assembly task is extended to a larger geometric scale, the object-based task constraints and the human hand-based task constraints should usually remain the same. However, the location-based task constraints for the extended task must be updated to fit the extended tasks.

In the two-dimensional assembly scenarios, each assembly location in the human-demonstrated tasks is regarded as a center of a cluster and then each assembly location in human-demonstrated tasks can be mapped to a new center of a cluster in the extend tasks throughout a linear scaling transformation, which can be written as

$$x' = \frac{W_{ex}}{W_d} x, \quad y' = \frac{W_{ex}}{W_d} y \tag{20}$$

where W_{ex} and H_{ex} are the width and height of the enlarged assembly process. W_d and H_d are the width and height of the demonstrated assembly. The location-based task constraint at each assembly location (x,y) in human-demonstrated tasks formats a center of location-based constraint cluster at (x',y') in the enlarged assembly task. Afterward, the constraint of each assembly location in the extended assembly task is determined by the k-nearest neighbor (KNN) classifier, which can be written as

$$c_{ex}(x_{ex}, y_{ex}) = C^{KNN}_{H_d \times W_d}(x', y') \tag{21}$$

where $x_{ex} \in [1, W_{ex}]$ and $y_{ex} \in [1, H_{ex}]$ gives a specific assembly location in the extended assembly task, c_{ex} is the location-based task constraint corresponding to the given assembly location. The right side of the equation means selecting the same location-based task constraint corresponding to the center of the closest cluster among all the nearby clusters which are centered at different (x',y'). The minimal Euler distance is used as the criterion for the KNN to select the closest cluster in the enlarged assembly task.

When the collaborative assembly task is extended to a larger scale, the MDP model in TC-IRL must also be updated to fit the extended tasks. After knowing the dimensions (height and the width) of the extended task from the human, the task constraints will be first updated for the new task. Based on the new task constraints, the task constraint-guided states, actions, and rewards can be defined in the same way as the originally demonstrated task in Section III.B. to construct a new task constraint-guided MDP and therefore result in a new TC-IRL. The new MDP and TC-IRL will also retain the advantage of small size for state and action space because the learned task constraint has been fully extended to the new scalable tasks to guide the definition and state and action space.

Because the human does not change for the new scalable task, his or her preference on how to accomplish the task giving the state and action space should retain the same. Therefore, the feature functions will remain the same as the originally TC-IRL, and more importantly, the optimized weights W^* which are previously learned from human demonstrations can also be used for the new task. This means that we do not need to re-train the TC-IRL at all although the tasks have been extended to a larger scale. We only need to use the learned reward function to update the reward map for the new tasks with newly updated state and action space and then use the assistance action generation approach which is introduced in the following section to generate appropriate robot assistance for the human.

B. Robot Assistance Generation

After the reward map is updated based on the reward function for the new task, the value function of the extended MDP can be determined through value iterations:

$$V_{i+1}(s) = \max_{a}\left(\sum_{s'} R(s) + \gamma V_i(s')\right) \quad s.t. \quad s \xrightarrow{a} s' \tag{22}$$

where s' is the next state of the system after the action a is executed at the state s. The converged value function with respect to the state s is noted as $V(s)$.

Since we aim to make the robot assist human to accomplish the assembly task, we will, therefore, require the human to initialize the task. The human first needs to accomplish two assembly actions based on his/her preference to establish an initial condition for the robot. Then, to generate appropriate assistance, the robot will first recognize the current state of the task via its sensing system in real-time and then determines which action to choose from the action space of the extended MDP model.

$$a^* = \arg\max_{a}\left(R(s) + \gamma V(s')\right) \quad s.t. \quad s \xrightarrow{a} s' \tag{23}$$

The optimized action a^* can also infers the assembly location that the human should be working on, the robot can then search all the available parts that can be assembled to this assembly location based on the location-based constraints. To determine which robot arm should be used to pick which available part, the Euclidean distances from the work home positions of both robot arms to each available part are calculated. The arm and the part corresponding to the minimal Euclidean distance are paired. According to the selected part, the robot then generates the list of all the available tools based on the object-based constraints. Similarly, the arm and the tool corresponding to the minimal Euclidean distance are paired. Based on the selected part-tool pair, the delivery targets are determined with the human hand preference. Afterward, the robot will execute the pick and handover actions to use different arms to pick up the correct part and tool and deliver them to the correct hands of the human to assist him/her to accomplish the assembly task.

V. Experimental Results and Analysis

A. Experiment Setup

The human stands face to face with the ABB Yumi robot to work in a shared workspace. The six cameras of the VICON motion capture system are set up on the roof surrounding the workspace to capture the motion of human hands. The origin of VICON is set at the center of the desk, and the location of each grid is calibrated. The Kinect RGB-D sensor offers a top-view point cloud of the workspace, which is used to verify the object states besides the VICON system.

In our experiments, the configuration of the workspace is illustrated in Fig. 1. The part/fastener stack zone is close to the robot, and the assembly zone is near to the human in the shared workspace. The 15 fasteners belong to three different types, square, hex, and crisscross. The 18 parts belong to three different types, square, hex, and crisscross, and three different colors, red, yellow, and blue. Mechanically, a part and a fastener can be assembled if and only if they have the same type. Initially, all the parts and fasteners are sorted in the stack zone and the assembly zone is empty.

In the human demonstration process, the human manipulates the parts and fasteners in the workspace by his/her both hands directly. The human operations are tracked by hand motion capture. Since the object-based constraints and the human hand preference are learned via statistic-based approaches, the sample size is critical for the human teaching and robot learning phase. The human starts the demonstration according to the natural language introduction of the robot. After the human finishes all the object manipulations through naturally pick-assembly-place operations, the human should put both hands at the work home position to indicate the robot

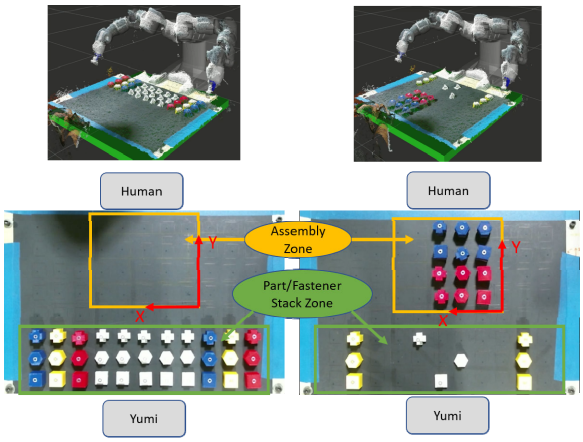

Fig. 1. The workspace configuration and collaborative assembly

TABLE I.
RESULTS OF ROBOT ASSISTANCE IN 4 X 3 ASSEMBLY

State	Process Prediction	Robot Assistance		
		Arm	Part/Tool	Hand
[10, "000000000011"]	9 (1, 4)	L	Part: B/H	R
		L	Tool: W/H	L
[9, "000000000111"]	8 (3, 3)	L	Part: B/S	R
		R	Tool: W/S	L
[8, "000000001111"]	7 (2, 3)	R	Part: B/C	R
		L	Tool: W/C	L
[7, "000000011111"]	6 (1, 3)	R	Part: B/S	R
		L	Tool: W/S	L
[6, "000000111111"]	5 (3, 2)	R	Part: R/C	R
		R	Tool: W/C	L
[5, "000001111111"]	4 (2, 2)	R	Part: R/H	R
		R	Tool: H	L
[4, "000011111111"]	3 (1, 2)	R	Part: R/S	R
		L	Tool: W/S	L
[3, "000111111111"]	2 (3, 1)	L	Part: R/C	R
		R	Tool: W/C	L
[2, "001111111111"]	1 (2, 1)	L	Part: R/H	R
		L	Tool: W/H	L
[1, "011111111111"]	0 (1, 1)	L	Part: R/S	R
		R	Tool: W/S	L
[0, "111111111111"]	-1 (end)	Stop	N/A	N/A

* Arm: L-pick by left robot arm, R- pick right robot arm.
* Part: B-blue, R-red, H-hex, S-square, C-crisscross.
* Tool: W-white, H-hex, S-square, C-crisscross.

that the demonstration is accomplished.

B. Results of TC-IRL in Collaborative Assembly

In the experiment, the proposed approaches are verified on a designed assembly task that the upper half of the assembly locations are expected to be red parts but no specific constraint in shape; the lower half of the assembly locations are expected to be blue parts but no constraint on the shape. Three rounds of 2 x 2 human demonstrations are given. In the demonstrations, the parts in red color but with different shapes are installed to the assembly locations in the upper half plane, while the parts in blue color but with different shapes are installed to the assembly location in the lower half plane. Meanwhile, human always use the left hand for tool operations and manipulate the parts with the right hand. All the three demonstrations are accomplished raw by raw, right to left, and from near to far with respect to the human's body position. The overall process of the 2 x 2 human demonstration is shown in the attached video.

The result of a collaborative assembly case for an extended 4 x 3 assembly task based on the human demonstrations of the previous section is shown in Table I.. In the human-robot collaboration phase, the human first picked up the blue/crisscross with his right hand and a corresponding white/cross tool with his left hand and placed the assembled part at location 11 (3, 4). Then, the human picked up a blue/hex part with his right hand and a corresponding white/hex tool with his left hand and placed the assembled part at location 10 (2, 4). After this, the task state was successfully initialized by the human, and the robot started to assist the human in the following steps of the task. The column of process prediction in Table I. gives the assembly locations predicted by the robot at different task states. The columns of robot assistance illustrate the robot's decisions on using which robot arm to pick up which part or tool and delivering to which human hand based on the real-time task state and the learned task constraints. The overall process of the robot assistance is shown in the attached video.

C. Evaluations

To evaluate the proposed model, we tested the model with different assembly processes, human hand preferences, and extended tasks with larger dimensions in assembly with 9 different kinds of parts and 3 different kinds of tools. Each participant is first introduced about how to demonstrate the object-based constraint, location-based constraint, human hand-based constraint, and assembly process by giving three human demonstrations with 2 x 2 dimensions based on his/her preferences. After the robot learning phase, the 2 x 5 and 4 x 3 collaborative assembly tasks are accomplished with robot assistance. The results in Table III show that with correctly calibrated the motion capture system, the robot tool center points, and the location of the objects, the proposed approach can obtain 100% accuracy in assembly sequence prediction, pick-delivery action

Based on the same workspace configuration and the same collaborative assembly task, the comparison on the action space size, the state space size, the transition map size, and computation effort are shown in Table II. For the TC-IRL, the corresponding results are automatically generated online by setting the proper parameters of the task size. For conventional IRL, we assume that the robot can use either left or right arm to pick up a part or a tool then deliver to either left or right human hand in each manipulation. The sizes of state and action spaces are then calculated for conventional IRL respectively.

We can see that with TC-IRL, the size of the action space, state space and state transition matrix of the MDP process in the model are significantly reduced compared with the model without the constraint extractions. Based on our task

TABLE III
Statistic Results of Robot Assistance

Assembly Process	Hand Preference	No. of Prediction (2x5)	No. of Pick (2x5)	No. of Handover (2x5)	No. of Prediction (4x3)	No. of Pick (4x3)	No. of Handover (4x3)	Accuracy
1	L	8/8	16/16	16/16	10/10	20/20	20/20	100%
	R	8/8	16/16	16/16	10/10	20/20	20/20	100%
2	L	8/8	16/16	16/16	10/10	20/20	20/20	100%
	R	8/8	16/16	16/16	10/10	20/20	20/20	100%
3	L	8/8	16/16	16/16	10/10	20/20	20/20	100%
	R	8/8	16/16	16/16	10/10	20/20	20/20	100%
4	L	8/8	16/16	16/16	10/10	20/20	20/20	100%
	R	8/8	16/16	16/16	10/10	20/20	20/20	100%

* The elements from the 2nd column to the 7th column are represented by "number of correct actions / numbers of total actions"

TABLE II
TC-IRL vs Conventional IRL

Task Size	Size of Action Set		Size of State Set		Size of Transition Map		Computation Effort	
	TC-IRL	IRL	TC-IRL	IRL	TC-IRL	IRL	TC-IRL	IRL
2 x 2	4	432	28	1.2754e7	3136	7.0277e16	0.00076 s	>10 min
2 x 3	6	648	186	2.7894e11	207576	5.0420e25	0.02174 s	>10 min
2 x 4	8	864	1016	1.1387e16	8258048	1.1204e35	0.4286 s	>10 min
2 x 5	10	1080	5110	7.4713e20	261121000	6.0287e44	11.10 s	>10 min
4 x 3	12	1296	24564	7.1895e25	7240681152	6.6989e54	292.2 s	>10 min

* The 2 x 2 tasks are used in human demonstrations on the real robot
* The 2 x 5 and 4 x 3 tasks are tested on the real robot
* The computation effort includes the time cost of the extended MDP generation and training of IRL

configuration, the size of the state and action space for TC-IRL is significantly smaller than conventional IRL and the advantages become more obvious when the task dimension increase, which leads to a dramatic increase of the state and action space for IRL. Based on the principle of IRL, The reduced size also implies reduced requirement on the training data, which is why our proposed approach only requires several human demonstrations. At the same time, because of the reduced size, the computational cost is also significantly reduced, which leads to better real-time performance.

VI. Conclusions

In this paper, we proposed a new learn-to-collaboration approach with TC-IRL method that generate robot assistance to assist humans in human-robot collaborative assembly. The task constraint guided IRL approach can significantly reduce the size of the action and state space and lead to a reduced requirement of training data and computational cost compared to traditional IRL. The proposed approach can also allow humans to teach the robot to accomplish new larger-scale tasks by learning from several small-scale demonstrations. The experiment results demonstrated the effectiveness and advantages of the proposed approach. The proposed approaches can potentially integrate into existing robot systems and workflow if the robot systems are collaborative and have proper vision or other sensing system to indicate the human operations. The objects used in the experiment can be replaced by real industrial parts with proper definitions of the tasks.

VII. Acknowledgements

This work was supported by the National Science Foundation under Grant IIS-1845779.

References

[1] P. Tsarouchi, S. Makris, and G. Chryssolouris, "Human–robot interaction review and challenges on task planning and programming," *Int. J. Comput. Integr. Manuf.*, vol. 29, no. 8, pp. 916–931, Aug. 2016.

[2] S. Calinon, F. Guenter, and A. Billard, "On learning the statistical representation of a task and generalizing it to various contexts," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pp. 2978–2983.

[3] M. Riley, A. Ude, and C. G. Atkeson, "Methods for motion generation and interaction with a humanoid robot: Case studies of dancing and catching," Georgia Institute of Technology, 2000.

[4] K. Kukliński *et al.*, "Teleoperation for learning by demonstration: Data glove versus object manipulation for intuitive robot control," in *2014 6th international congress on Ultra modern telecommunications and control systems and workshops (ICUMT)*, 2014, pp. 346–351.

[5] F. J. Abu-Dakka, B. Nemec, A. Kramberger, A. G. Buch, N. Krüger, and A. Ude, "Solving peg-in-hole tasks by human demonstration and exception strategies," *Ind. Robot An Int. J.*, 2014.

[6] M. Edmonds *et al.*, "Feeling the force: Integrating force and pose for fluent discovery through imitation learning to open medicine bottles," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 3530–3537.

[7] M. Ferreira, P. Costa, L. Rocha, and A. P. Moreira, "Stereo-based real-time 6-DoF work tool tracking for robot programing by demonstration," *Int. J. Adv. Manuf. Technol.*, vol. 85, no. 1–4, pp. 57–69, 2016.

[8] A. Vakanski, I. Mantegh, A. Irish, and F. Janabi-Sharifi, "Trajectory Learning for Robot Programming by Demonstration Using Hidden Markov Model and Dynamic Time Warping," *IEEE Trans. Syst. Man. Cybern.*, vol. 42, no. 4, pp. 1039–1052, 2012.

[9] L. Rozo, P. Jiménez, and C. Torras, "Force-based robot learning of pouring skills using parametric hidden Markov models," *9th Int. Work. Robot Motion Control. RoMoCo 2013 - Work. Proc.*, pp. 227–232, 2013.

[10] H. Goto, J. Miura, and J. Sugiyama, "Human-Robot Collaborative Assembly by On-line Human Action Recognition Based on an FSM Task Model," *Int. Conf. Human-Robot Interact. - Work. Collab. Manip.*, pp. 1–6, 2013.

[11] Y. Wang, R. Xiong, L. Shen, K. Sun, J. Zhang, and L. Qi, "Towards learning from demonstration system for parts assembly: A graph based representation for knowledge," *4th Annu. IEEE Int. Conf. Cyber Technol. Autom. Control Intell. Syst. IEEE-CYBER 2014*, pp. 174–179, 2014.

[12] S. Kim, R. Haschke, and H. Ritter, "Gaussian Mixture Model for 3-DoF orientations," *Rob. Auton. Syst.*, vol. 87, pp. 28–37, 2017.

[13] W. Wang, R. Li, Y. Chen, Z. M. Diekel, and Y. Jia, "Facilitating Human-Robot Collaborative Tasks by Teaching-Learning-Collaboration From Human Demonstrations," *IEEE Trans. Autom. Sci. Eng.*, no. 99, pp. 1–14, 2018.

[14] S. Nikolaidis, R. Ramakrishnan, K. Gu, and J. Shah, "Efficient model learning from joint-action demonstrations for human-robot collaborative tasks," in *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, 2015, pp. 189–196.

[15] D. Malik, M. Palaniappan, J. F. Fisac, D. Hadfield-Menell, S. Russell, and A. D. Dragan, "An efficient, generalized bellman update for cooperative inverse reinforcement learning," *35th Int. Conf. Mach. Learn. ICML 2018*, vol. 8, pp. 5435–5443, 2018.

[16] U. Kartoun, H. Stern, and Y. Edan, "A Human-Robot Collaborative Reinforcement Learning Algorithm," *J. Intell. Robot. Syst.*, vol. 60, no. 2, pp. 217–239, 2010.

[17] M. K. Holder, "Why are more people right-handed," *Sci. Am. Inc*, 1997.

[18] M. Wulfmeier, P. Ondruska, and I. Posner, "Maximum Entropy Deep Inverse Reinforcement Learning," 2015.