Integrating Uncertainty-Aware Human Motion Prediction Into Graph-Based Manipulator Motion Planning

Wansong Liu, Kareem Eltouny, Sibo Tian, Xiao Liang, and Minghui Zheng

***Abstract*—There has been a growing utilization of industrial robots as complementary collaborators for human workers in remanufacturing sites. Such a human–robot collaboration (HRC) aims to assist human workers in improving the flexibility and efficiency of labor-intensive tasks. In this article, we propose a human-aware motion planning framework for HRC to effectively compute collision-free motions for manipulators when conducting collaborative tasks with humans. We employ a neural human motion prediction model to enable proactive planning for manipulators. Particularly, rather than blindly trusting and utilizing predicted human trajectories in the manipulator planning, we quantify uncertainties of the neural prediction model to further ensure human safety. Moreover, we integrate the uncertainty-aware prediction into a graph that captures key workspace elements and illustrates their interconnections. Then, a graph neural network (GNN) is leveraged to operate on the constructed graph. Consequently, robot motion planning considers both the dependencies among all the elements in the workspace and the potential influence of future movements of human workers. We experimentally validate the proposed planning framework using a 6-degree-of-freedom manipulator in a shared workspace where a human is performing disassembling tasks. The results demonstrate the benefits of our approach in terms of improving the smoothness and safety of HRC. A brief video introduction of this work is available as the supplemental materials.**

Manuscript received 20 January 2024; revised 6 April 2024; accepted 10 May 2024. Date of publication 5 June 2024; date of current version 16 August 2024. Recommended by Technical Editor T. Shimono and Senior Editor Q. Zou. This work was supported by USA National Science Foundation under Grant 2026533/2422826 and Grant 2132923/2422640. *(Corresponding authors: Minghui Zheng; Xiao Liang.)*

This work involved human subjects or animals in its research. The author(s) confirm(s) that all human/animal subject research procedures and protocols are exempt from review board approval.

Wansong Liu is with the Mechanical and Aerospace Engineering Department, University at Buffalo, Buffalo, NY 14260 USA (e-mail: wansongl@buffalo.edu).

Kareem Eltouny is with the Civil, Structural and Environmental Engineering Department, University at Buffalo, Buffalo, NY 14260 USA (e-mail: keltouny@buffalo.edu).

Sibo Tian and Minghui Zheng are with the J. Mike Walker '66 Department of Mechanical Engineering, Texas A&M University, College Station, TX 77840 USA (e-mail: sibotian@tamu.edu; mhzheng@tamu.edu).

Xiao Liang is with the Zachry Department of Civil & Environmental Engineering, Texas A&M University, College Station, TX 77840 USA (e-mail: xliang@tamu.edu).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TMECH.2024.3402682.

Digital Object Identifier 10.1109/TMECH.2024.3402682

***Index Terms*—Graph neural network, human motion prediction, motion planning.**

I. Introduction

TO FACILITATE efficient and safe disassembly, robots are usually employed as complementary collaborators to work closely with human operators [1], [2]. In such close collaboration, robots are required to generate collision-free motions and adjust their motions efficiently. The planning problem turns out to be complicated when human operator's behaviors are involved since real-time responsiveness necessitates quick motion generation in the constantly changing configuration space. Manipulators must respond adaptively to human operator's actions. Predicting human motions allows collaborative robots to proactively plan motions, ensuring a safe and seamless human–robot collaboration (HRC) [3].

Integrating human motion prediction into robotic motion planning has two technical challenges. One is that human motion is inherently complex and stochastic, which requires robust prediction models to handle the uncertainties arising from human behavior variations or unexpected actions [4], [5], [6]. Addressing such uncertainties holds particular importance in terms of ensuring safety of HRC as unreliable predictions can potentially result in the planning of a dangerous trajectory. The second challenge is that the prediction as well as the uncertainty introduce additional computational complexity for generating robot motions [7]. To have a real-time responsiveness in HRC, the planning algorithms must integrate the prediction and the uncertainty in an efficient way and find collision-free motions within tight time constraints.

In this article, we propose a motion planning framework to enhance the seamless and safe collaboration between humans and manipulators in disassembly processes. Fig. 1 shows the overview of the framework. The framework is comprised of two modules. The first one is the uncertainty-aware human motion prediction. It seeks to provide future trajectories of human operators and the uncertainty of the network-based prediction model for the purpose of safe manipulator motion planning. The second module is a graph-based neural motion planner that incorporates uncertainty-aware prediction and generates collision-free manipulator motions. We transform the collaboration workspace into a graph representation that encapsulates the relationships and dependencies among the objects within the

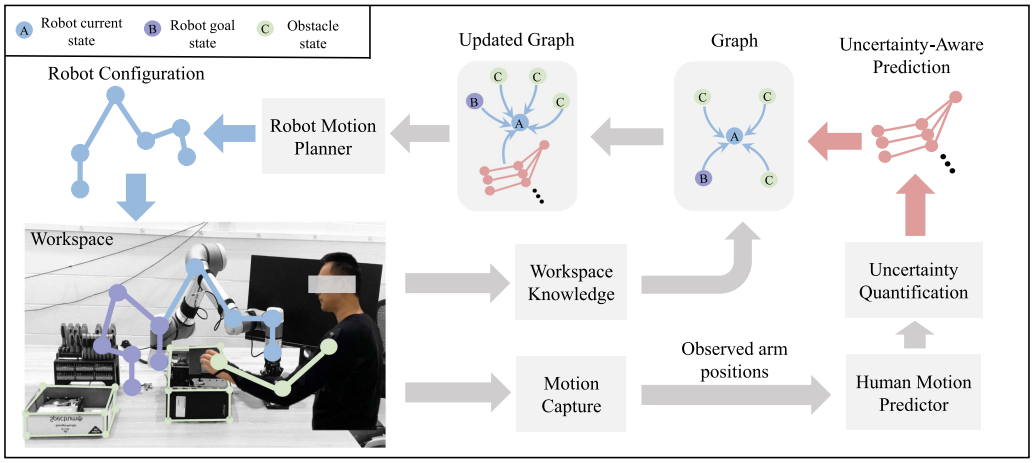


Fig. 1. Overview of the proposed HRC motion planning framework: Observed human motions are used to compute future uncertainty-aware human motions; collaboration workspace is converted to a graph. It preserves the structural attributes of objects by employing multiple nodes and edges. The key elements and characteristics of the workspace are depicted through nodes with features, and their connections are established through edges. Note that we only show a few of the nodes in the figure to simplify the illustration. Uncertainty-aware prediction is also represented using nodes and edges and naturally integrated into the overall graph. GNN-based motion planner eventually generates a safe robot configuration, directing the robot toward the desired goal, while avoiding the moving human agent in close vicinity.

workspace. The uncertainty-aware predictions are represented as nodes and edges, which are intuitively integrated into the constructed graph and interconnected with other objects.

In summary, the main contributions of this work are summarized as follows.

1) We present a framework for HRC that naturally integrates motion planning of high-DOF robot manipulators with uncertainty-aware human motion prediction, using graph neural networks (GNNs).
2) The inherent uncertainty of the human motion prediction model is incorporated into robot motion planning intuitively and conveniently (i.e., using nodes and edges) to enhance safety of HRC.
3) We conduct comprehensive experimental studies within a collaborative disassembly scenario to validate the performance of our model. The proposed planning framework showcases the benefits in terms of earlier robot's response and near-optimal trajectory planning when a sudden human intervention occurs.

II. Related Works

A. Human Motion Prediction

Traditional statistic-based models have been utilized to learn the probability distribution of human motion, enabling them to reason about possible future human trajectories based on historical data, such as the hidden Markov model [8] and the Gaussian regression model [9]. Although these probabilistic methods are suited for capturing the stochastic nature of human motion, their performance tends to be less satisfactory when dealing with intricate motion patterns. To predict complex human motion, recurrent neural networks (RNNs) have been widely used to obtain deterministic future human trajectories [10]. In addition, graph convolutional networks [11], [12] and Transformer [13], [14] have recently become popular in human motion prediction. These works show significant improvement in capturing the spatial and temporal dependencies of human motion data.

Instead of blindly trusting the predicted human motions, existing studies quantified the uncertainty of the predicted human motions using statistic-based prediction models, e.g., [4], [9], [15], and [16]. These models can naturally predict trajectories in a probabilistic way, handling irregular human movements in HRC. While network-based models typically provide deterministic predictions, some studies have developed techniques to measure the uncertainty inherent in these models and to provide the confidence level associated with the model's outputs. For example, Cheng et al. [17] developed a parameter-adaption-based neural network to provide uncertainty bounds of the prediction in real time. Zhang et al. [18] employed conditional variational autoencoders to sample multiple saliency maps from the latent space, ultimately obtaining an accurate saliency map using the quantified uncertainty. Eltouny et al. [6] trained an ensemble of motion prediction network models, and estimated the uncertainty based on the aggregation of diverse motion predictions.

B. Robot Motion Planning

One of the most important problems in HRC is to plan collision-free robot motions in dynamic workspaces. The computational expense imposed by the curse of dimensionality limits the application of traditional grid-based methods, such as A* algorithm [19]. Random-sampling-based methods, such as the rapidly exploring random tree (RRT) [20], have demonstrated effectiveness in high-dimensional planning problems. Furthermore, to ensure the optimality of robot trajectories, asymptotically optimal sampling-based such as batch-informed trees [21], fast marching trees (FMT*) [22], and optimization-based methods [23], [24] are developed. Nowadays, network-based motion planners have been widely used to generate near-optimal robot trajectories with low computational cost. For example, the work in [25], [26], and [27] leveraged a network-based model to

imitate expert robot trajectories generated from oracle planners, providing near-optimal robot motions. Furthermore, rather than imitating expert trajectories, the work in [28] and [29] employed reinforcement learning to obtain optimal policies to generate robot motions. Despite the advantages demonstrated by such planners, they may struggle to capture the intrinsic connectivity among objects within the workspace. Rather than blindly preprocessing all data together, the graph representation proposed in [30] highlights the both local and global dependencies of objects in the workspace when generating robot motions, which however does not explicitly consider human motion prediction in the motion planner.

Incorporating human motion prediction into robotic planning can improve the efficiency of HRC. Cheng et al. [31] included the task recognition and trajectory prediction of human workers into HRC systems to significantly improve efficiency. Unhelkar et al. [7] proposed a planning algorithm that leverages the prediction of nearby humans to efficiently execute collaborative assembly tasks. Moreover, incorporating the prediction is beneficial for generating collision-free robot trajectories proactively, thus expanding the safety margin of the collaboration. Park et al. [9] used the predicted human motion to compute collision probabilities for safe motion planning. Kratzer et al. [32] proposed a prediction framework that enables the mobile robot to avoid the possible area occupied by a human partner. Zheng et al. [33] developed an encoder–decoder network to predict human hand trajectories, and integrated the avoidance of future collisions as constraints into a model predictive control framework, allowing the planning of safe trajectories.

III. Uncertainty-Aware Human Motion Prediction

In this section, we introduce an RNN-based human motion prediction model and explain how the uncertainty of the model is quantified.

A. Human Motion Predictor

To predict human trajectories during task execution, we train a prediction model based on an RNN with long short-term memory (LSTM) architecture. Rather than using 3-D position data of arm joints, we employ unit vectors of bones for the network training. In this case, we can ensure a consistent distance between two joints when reconstructing arm poses from bone vectors using the corresponding bone lengths. This choice preserves the anatomical constraints of the arm during the prediction process. The human arm bone vector is denoted as $x = (\phi_1, \phi_2) \in \mathbb{R}^6$, where $\phi_1 \in \mathbb{R}^3$ and $\phi_2 \in \mathbb{R}^3$ are two bone vectors of human upper-arm and forearm, respectively. Notably, the position of arm joints and human arm occupied area in the workspace can be reconstructed using x and anthropometric parameters p_h, which contain the average bone length and radius of the human arm for each segment. The prediction process is denoted as

$$\hat{X} = F(X, \mathbf{W}) \tag{1}$$

where $X{=}[x_{-N+1}, \ldots, x_0] \in \mathbb{R}^{6N}$ is the human motion of observed N steps, $F(\bullet)$ indicates the prediction function, and $\hat{X}{=}[\hat{x}_1, \ldots, \hat{x}_m, \ldots, \hat{x}_M] \in \mathbb{R}^{6M}$ stands for the human motion of predicted M steps. In addition, we treat the well-trained network as the prediction model, and $\mathbf{W}$ indicates the learning weights of the network after training.

B. Uncertainty Quantification Using Monte Carlo Dropout Sampling (MCDS)

The previous section briefly introduces that using a network-based motion predictor can predict human trajectories in upcoming time steps. However, human motions in HRC exhibit a certain level of variability that is influenced by factors, such as individual characteristics and worker fatigue. Therefore, it is necessary to explicitly quantify uncertainties, enabling effective consideration of variations in human movements. We employ MCDS to quantify the uncertainty of our prediction model, considering that it provides accurate uncertainty estimations and only requires training a single model.

We aim to obtain the prediction distribution such that the possible future trajectories can be utilized for safe robot motion planning. To this end, we apply dropout to every layer of the prediction model, and treat it as a Bayesian approximation of a Gaussian process model over the prediction model parameters [34]. The prediction distribution is calculated using the following equation:

$$p\left(\hat{X}|X\right) = \frac{p\left(\hat{X}|X, \mathbf{W}\right) p(\mathbf{W})}{p\left(\mathbf{W}|X, \hat{X}\right)} \tag{2}$$

where $p(\mathbf{W})$ is a prior Gaussian distribution over the model parameters, $p(\hat{X}|X, \mathbf{W})$ indicates the likelihood used to capture the prediction process, and $p(\mathbf{W}|X, \hat{X})$ denotes the posterior distribution.

Considering that the posterior distribution can not be evaluated analytically, we use variational inference to approximate it. The approximating distribution $q(\mathbf{W})$ can be close to the true posterior distribution by minimizing the Kullback–Leibler (KL) divergence between them

$$\mathrm{KL}\left(q(\mathbf{W}) \,||\, p\left(\mathbf{W}|X, \hat{X}\right)\right) \tag{3}$$

where $q(\mathbf{W})$ is defined using Bernoulli distributed random variables and some variational parameters that can be optimized. As pointed out in [35], the training of the prediction model would also be beneficial for minimizing the KL divergence term. Therefore, $q(\mathbf{W})$ is optimized after the network training, and sampling from $q(\mathbf{W})$ is equivalent to applying dropout on each layer of the prediction model. Eventually, the predictive variance u at test time is calculated using the following equation:

$$\mathbf{u} \approx \frac{1}{K-1} \sum_{k=1}^{K} \left[F\left(X, \overline{\mathbf{W}}_k\right)^T F\left(X, \overline{\mathbf{W}}_k\right) - KE^T E\right] \tag{4}$$

where $u = [u_1, \ldots, u_m, \ldots, u_M]$ indicates the prediction variance, K is the Monte Carlo sampling size, $\overline{\mathbf{W}}_k$ is fitted to $q(\mathbf{W})$ and denotes the model parameters of the kth sample, and $E \approx \frac{1}{K}\sum_{k=1}^{K} F(X, \overline{\mathbf{W}}_k)$ represents the predictive mean.

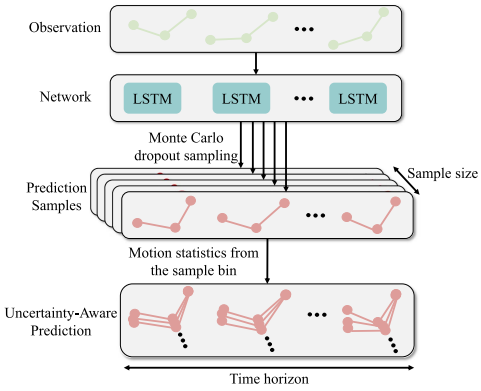


Fig. 2. Uncertainty quantification of the prediction model: Green dots are the observed human joints, red dots are the predicted human joints, and the uncertainty-aware prediction is generated based on the predictive distribution and includes multiple possible human arm poses at each time step.

The process of obtaining uncertainty-aware human motion prediction is illustrated in Fig. 2. The observed human motion is propagated into a well-trained LSTM model. MCDS is employed to generate different possible configurations of the network parameters, and multiple prediction samples are obtained. Finally, the uncertainty-aware prediction includes multiple possible human arm poses at each time step, and is denoted as $\hat{X}^*=[\hat{x}_1^*,\ldots,\hat{x}_m^*,\ldots,\hat{x}_M^*]\in\mathbb{R}^{6M\times K}$. Notably, we use $*$ to indicate there are multiple possible arm poses at a predicted time instance, and these poses fit a normal distribution $*\sim\mathcal{N}(E,u)$, where E represents the mean and u indicates the predictive variance.

IV. Graph-Based Motion Planner

This section presents 1) explanations of converting the collaboration workspace and the uncertainty-aware prediction to a graph representation; 2) details of how to leverage a GNN to operate on the constructed graph and generate near-optimal robot motions.

A. Graph Representation: Illustrating Features and Connections of Objects in the Workspace

Rather than simply imitating reference motions like traditional neural motion planners, our approach aims to emphasize the dependencies of each key object within the workspace since the object dependencies significantly influence the planning of robot motions. To highlight such dependencies in the planning, we use nodes to represent essential objects in the collaboration workspace and connect them using edges. As shown in Fig. 3, the robot's current state is denoted using six blue nodes, corresponding to six joints of the robot. The same representation strategy is applied to the robot's goal state and the obstacle states. To simplify the illustration, we respectively use dots A, B, and C to represent the robot's current state, the robot's goal state, and the obstacle's state in the graph of Fig. 3. Furthermore, the uncertainty-aware prediction contains multiple future human arm joint positions, which are represented as nodes. In summary, we use v to represent the node of the graph, and

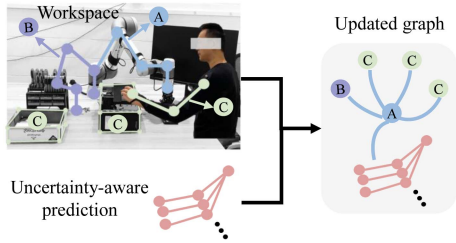


Fig. 3. Representation of objects in the workspace and the simplified illustration of the overall graph: the uncertainty-aware prediction is represented using red color. We use dots A, B, and C to simplify the graph representation. Blue dot A indicates the robot's current state, purple dot B denotes the robot's goal state, and green dot C indicates the obstacle's state. All dots contain multiple nodes and edges based on their own structural attributes.

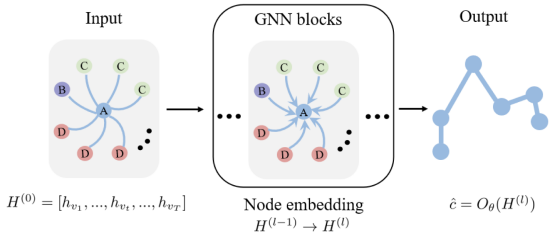


Fig. 4. Process of motion generation: red dot D indicates the predicted human arm after MCDS, multiple GNN blocks are used to update the embeddings of nodes, and GNN finally outputs the robot configuration of the next step.

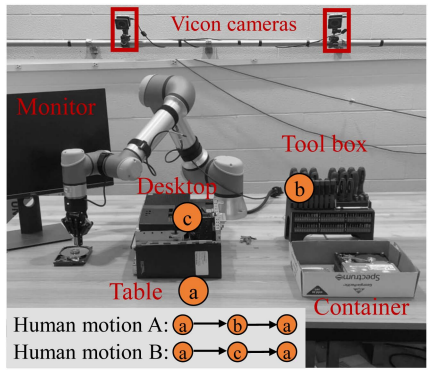


Fig. 5. Experimental platform: a, b, and c represent three locations in the table, tool box, and desktop, respectively. Human motion A demonstrates a scenario in which a human worker initially disassembles a hard disk on the table, then reaches towards the tool box to get a new screwdriver, and finally resumes the disassembly task. Human motion B demonstrates a scenario where a human worker initially disassembles a hard disk on the table, then grabs a component from the disassembled desktop, and eventually puts the component on the table.

$V=[v_1,\ldots,v_t,\ldots,v_T]$ stands for total T essential nodes in the collaboration workspace.

B. Graph Operation: Node Embedding Based on Neighbors

In the previous section, we employ a graphical representation to efficiently illustrate the objects in the collaborative workspace and showcase their connections. To generate collision-free robot

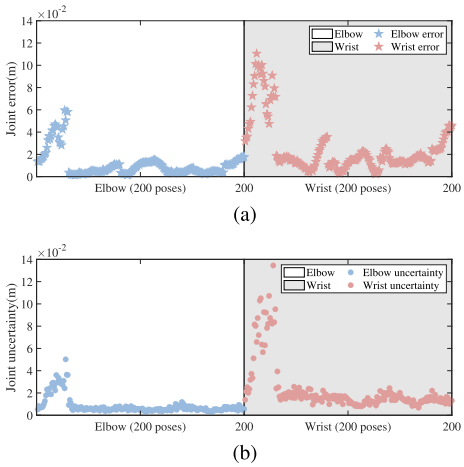


Fig. 6. Comparison between predicted errors and quantified uncertainties regarding elbow and wrist: High co-relation. (a) Predicted errors of elbow and wrist joint positions. (b) Quantified uncertainties in terms of elbow and wrist positions.

motions, we first employ an oracle planner that generates expert robot trajectories in collaborative workspaces to obtain the training data, and then leverage a GNN to operate on the constructed graph and train the network to generate near-optimal motions.

The graph is described by two matrices: a feature matrix and an adjacency matrix. The feature matrix H describes features of the objects in the workspace, such as the manipulator joint value, the current and future arm's positions, and the static obstacles' potions. The adjacency matrix A indicates the relationships between all nodes. The layers of GNN update features of each node based on the adjacency matrix A. The node embedding process is denoted as

$$h_{v_t}^{(l)} = f_{\text{update}}\left(\theta^{(l)}, h_{v_t}^{(l-1)}, \left\{h_{v_j}^{(l-1)}\right\}_{j\in\mathcal{N}_{v_t}}\right) \tag{5}$$

where $h_{v_t}^{(l)}$ denotes the embedding of node v_t in the layer l, θ is learning weights, and $\mathcal{N}_{v_t}$ indicates neighbors of node v_t.

Fig. 4 illustrates the motion generation using GNN. The red dot D indicates the uncertainty-aware predictions. The GNN input $H^{(0)} = [h_{v_1}, \ldots, h_{v_t}, \ldots, h_{v_T}]$ is initialized by the node features of the overall graph. All nodes update their embeddings simultaneously in the layer-wise propagation of GNN using the following equation:

$$H^{(l)} = \text{ReLU}\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l-1)}\Theta^{(l-1)}\right) \tag{6}$$

where ReLU is a nonlinear activation function, $H^{(l)}$ indicates the updated feature matrix in the layer l, $\tilde{A} = A + I$ is the adjacency matrix with an added identity matrix, $\tilde{D}$ stands for the degree matrix of $\tilde{A}$, and $\Theta^{(l-1)}$ is the learning weights in matrix form. Eventually, the output layer takes all updated node embeddings into account and provides the next robot configuration

$$\hat{c} = O_\theta\left(H^{(l)}\right) \tag{7}$$

where O_θ is the output function, and $\hat{c}$ is the next robot configuration toward the goal region.

C. GNN Training

The previous section describes that our neural planner considers various factors for generating the next robot configuration $\hat{c}$ toward the goal position. To ensure the generated motion to be near-optimal, GNN needs to be trained using optimal paths generated from an oracle planner. The optimal robot path connecting given start and goal configurations is denoted as $\sigma = [c_1, \ldots, c_i, \ldots c_I] \in \mathbb{R}^{I\times d}$, where d is the dimensionality of the robot configuration space. Utilizing the one-step look-ahead planning strategy outlined in [27], we define the training loss function for our neural planner as follows:

$$l_{\text{planner}} = \frac{1}{N_z}\sum_{z}^{N_z}\sum_{i=1}^{I_z-1}\|c_{z,i} - \hat{c}_{z,i}\|^2 \tag{8}$$

where N_z is the total number of robot paths in a batch, and I_z is the length of the zth path.

D. Bi-Directional Planning

After the network training, the learning weights Θ in (6) are well-tuned. We use the well-trained GNN to perform real-time robot motion planning. Based on the one-step-ahead planning strategy, the planned robot configurations are iteratively used as new inputs of our neural planner until a complete path is found. Such a planning heavily depends on the previously generated robot configuration, which may potentially accumulate errors throughout the planning process and result in the robot deviating from the goal region. Therefore, we adopt a bidirectional planning way to enhance the robustness of online planning.

The bidirectional planning starts by initiating two subplanning branches simultaneously, originating from the start and goal configurations, respectively. Then, we generate linear interpolations trying to directly connect two branches in each planning iteration. In addition, the two branches grow iteratively until the planned robot configurations and the generated interpolations are both collision-free. Eventually, the two subplanning branches are stitched together to be a complete robot path.

V. Experimental Validations

A. Experiment Setup

1) Experimental Setting: Fig. 5 shows the experiment setting of two disassembly scenarios. We use the Vicon motion capture system to track human motions. When constructing the human arm model for collision-checking, we introduce an additional radius to the human arm. This precautionary measure establishes a safety margin between the detected collision and the actual collision. To ensure safety and prevent potential physical injuries, robot needs to consider both the tracked and predicted human motions.

Note that in the collaboration scenario of this work, the human worker grabs tools located on the workstation while the robot transports disassembled components above the workstation. The physical barrier of the workstation effectively separates the entire human body from the robotic arm. Consequently, the human arm and the robot present the highest likelihood of collisions.

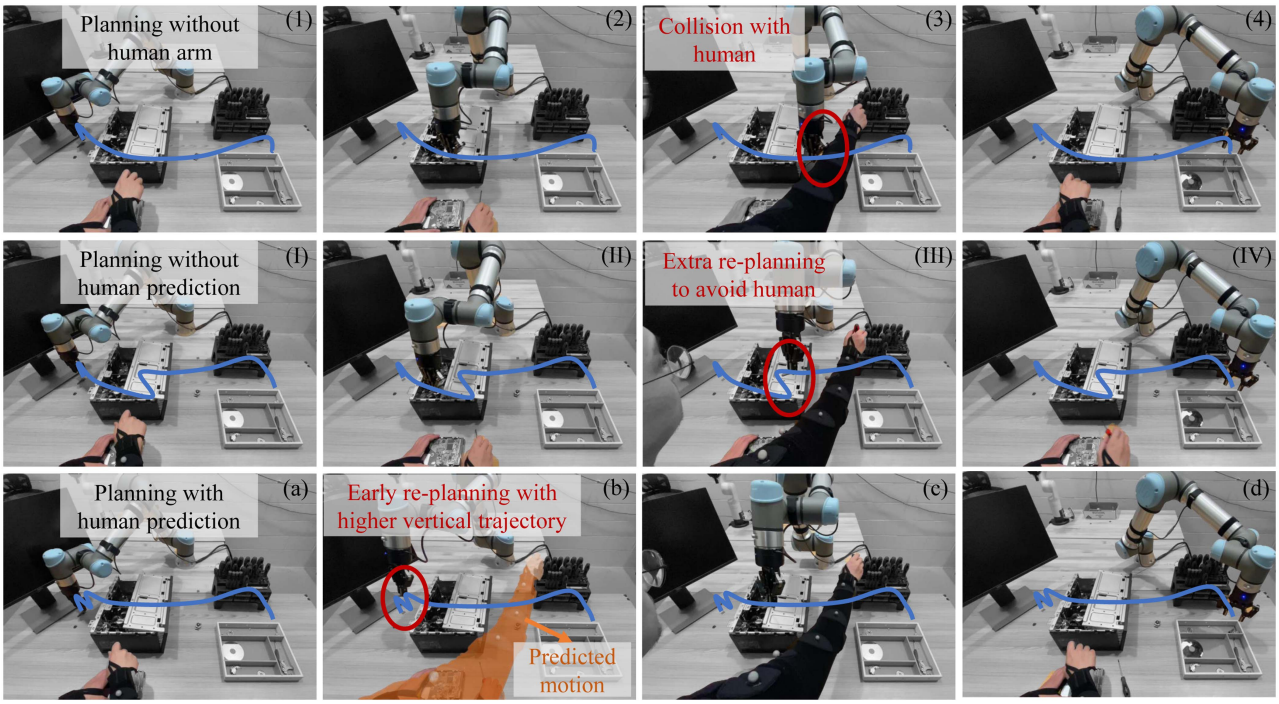


Fig. 7. Experimental tests based on human motion A: the experimental tests include three planning cases. (1)–(4) Planning scenario without considering the human arm. (I)–(IV) Planning based on the current human arm's position, where the robot replans its motions to accommodate the reaching motion of the human. (a)–(d) Planning with uncertainty-aware prediction. In subfigure (b), the observed human motion is putting the screwdriver on the table, and the predicted human motion is reaching for a new screwdriver (i.e., human arm represented by orange color). The robot detects potential future collisions based on the predicted human motion and has an early replanning to avoid such collisions. Note that the blue end-effector paths are drawn manually.

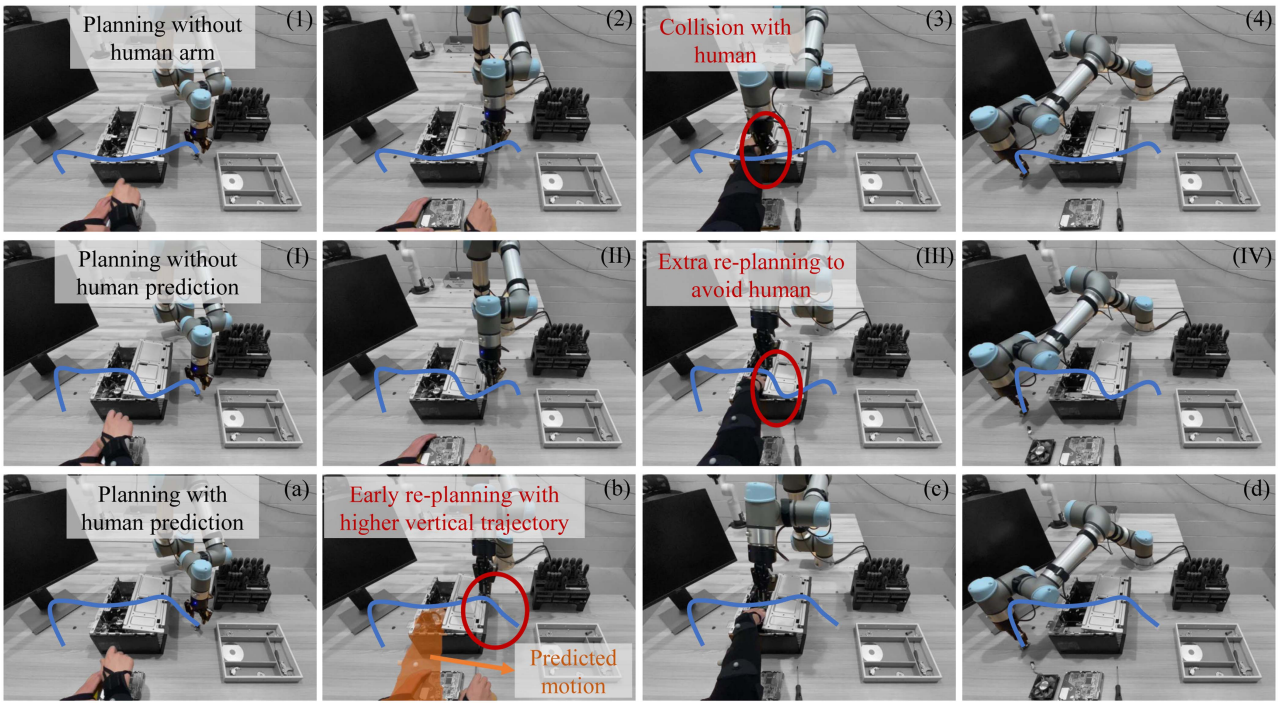


Fig. 8. Experimental tests based on human motion B: the experimental tests include three planning cases. (1)–(4) are the planning scenario without considering the human arm. (I)–(IV) Planning based on the current human arm's position, where the robot re-plans its motions to accommodate the grabbing motion of humans. (a)–(d) Planning with uncertainty-aware prediction. In subfigure (b), the observed human motion is putting the screwdriver on the table, and the predicted human motion is grabbing components from the disassembled desktop (i.e., human arm represented by orange color). The robot detects potential future collisions based on the predicted human motion, and has an early replanning to avoid such collisions. Note that the blue end-effector paths are drawn manually.

Therefore, tracking and predicting the positions of the forearm and upper-arm suffice to ensure the safety of the collaborative task as outlined in this work. However, it is inadequate for ensuring the safety in scenarios of broader collaboration. Tracking and predicting the movement of the whole human body instead of just the forearm and upper-arm are still necessary for diverse collaborative modes.

2) Data Acquisition: We collect 120 human arm trajectories in the frequency of 25 Hz for each type of human motion shown in Fig. 5. In our work, one human worker is involved in the data collection, and the human worker is required to perform actions naturally throughout task executions, without deliberate control over the speed. Therefore, while the action speed exhibits some variability, it remains within a reasonable range. These trajectories are then converted to bone-vectors for network training. We use 70% of data to train the prediction model. Another 15% of data are employed for validation, while the remaining portion is reserved for testing. The horizons of the observation and prediction are both 2 s.

3) Networks: The RNN-based human motion prediction model is based on LSTM structure. It consists of three LSTM layers and one dense layer serving as the output layer. Each LSTM layer is followed by a dropout layer with a dropout probability of 10%. The input dimension of the prediction model is a 50×6, where 50 indicates the observation steps and 6 implies the number features of arm poses. The output dimension of the model is $50 \times 5 \times 6$, where 50 implies the prediction steps, 5 indicates the sampling size, and 6 denotes the number features of the arm pose. The GNN consists of five graph convolutional layers employing rectified linear unit activation functions. Subsequently, one global sum pooling layer is applied to compute the sum of node features, and one dense layer is employed as the output layer. The input of the GNN is a constructed graph described with a 55×6 feature matrix and a 55×55 adjacency matrix, where 55 indicates the total node number and 6 implies the number of node features.

B. Experimental Test Results

1) Validation of the Neural Planner: To evaluate the effectiveness of the graph-based planner, we create a total of 12 different workspaces. We employ RRT* [36] from the open motion planning library (OMPL) as the oracle planner to generate optimal robot motions in a motion planning platform MoveIt. Each static workspace includes 800 planning scenarios with random pairs of start and goal configurations. In static workspaces, we conducted comparative studies between our approach and three other planners from OMPL, which are RRT* [36], RRT [20], and the advanced planner bidirectional FMT* (BFMT* [37]). A comparison study is provided in Table I [30], from which the graph planner demonstrates superior performance compared to the other three planners in terms of path length and planning time, and it achieves a promising level of success in generating collision-free motions.

2) Uncertainty-Aware Prediction: We select different values for Monte Carlo sampling size K to obtain the quantified uncertainty. The corresponding results are illustrated in Table. II. The "Elbow/m" and "Wrist/m" present the standard deviation in predictions relative to the mean predicted joint position. Small values of "Elbow/m" and "Wrist/m" indicate greater consistency among multiple arm poses at the predicted time instance, while larger values signify greater variability. Note that there are no target or minimum required values for the quantified uncertainties. A large value signifies increased variability in arm poses at the predicted time instance. This may broaden the scope of possible arm motions for enhanced robot motion planning. However, due to the close proximity between the potential human arm poses and the robot, it will make the robot more difficult to find motions to avoid potential collisions, and the prolonged inference time increases the risk of human–robot contacts. Therefore, the selection of suitable K is a tradeoff problem. Based on the observation of the Table. II, the rise in the value of K leads to a substantial increase in the inference time, whereas the escalation in the quantified uncertainties of elbow and wrist positions has a negligible impact. Therefore, we select $K = 5$ to quantify uncertainties and take the quantified uncertainties into the safe robot motion planning. This decision balances the need for comprehensive uncertainty assessment with inference time.

3) Comparison Between Predictive Error and Uncertainty: We also define the predictive error as the difference between the mean prediction and the ground truth, and compare the quantified uncertainties and predictive errors in terms of the human elbow and wrist joint positions in Fig. 6. It includes 200 arm poses selected from the test dataset. The predictive error and quantified uncertainty show a high co-relation. Importantly, when human workers are conducting collaborative tasks, the predictive error can not be utilized in the robot planning since it is not feasible to obtain future ground truth at the current time step. Therefore, the quantified uncertainty can be used as an alternative source of information for ensuring human safety.

TABLE I
COMPARISON RESULTS OF PLANNERS: WE ASSUME THE RESULT DATA FIT NORMAL DISTRIBUTION

Planner	Path length (m)	Planning time (s)	Success rate
RRT	1.695 ± 0.725	0.335 ± 0.650	93.5%
RRT*(±10%)	1.078 ± 0.138	3.700 ± 2.421	70.2%
BFMT*(±10%)	1.082 ± 0.137	0.541 ± 0.148	91.1%
Graph Planner	**1.023** ± **0.130**	**0.197** ± **0.053**	88.1%

Path length refers to the total distance traversed by the manipulator's end-effector along the planned path, planning time quantified the time taken by the planner to compute a collision-free path, and success rate measures the percentage of planning scenarios in which a collision free path is successfully generated from the given start to the goal configuration. Considering the dependency between planning time and trajectory optimality in RRT* and BFMT*, we terminate the planning of RRT* and BFMT* when the planned path length reaches a certain percentage of the path length generated by our approach. Note that the planning time results of RRT is imposed to be normal distribution for better comparison.

TABLE II
INFERENCE TIME AND THE STANDARD DEVIATION IN PREDICTIONS BASED ON DIFFERENT SAMPLING SIZES, WHERE THE VALUE IN THE PARENTHESES IS THE INCREASED PERCENTAGE COMPARED TO THE SELECTION OF $K = 5$

K	Elbow/m	Wrist/m	Inference time/s
5	5.25×10^{-3}	9.95×10^{-3}	0.29
10	5.69×10^{-3} (↑8.38%)	1.01×10^{-2} (↑1.50%)	0.49 (↑68.97%)
20	5.64×10^{-3} (↑7.43%)	1.01×10^{-2} (↑1.50%)	1.16 (↑300.00%)

TABLE III
QUANTITATIVE RESULTS OF THE SMOOTHNESS FOR THE ROBOT TRAJECTORY

Experimental Case	Acc. (rad/s^2)	Jerk. (rad/s^3)
Planning w/o human arm	1.71×10^{-2}	1.94×10^{-2}
Planning w/o human prediction	4.21×10^{-2}	10.1×10^{-1}
Planning w/ human prediction	3.50×10^{-2}	7.58×10^{-2}

The reported smoothness is evaluated by the average acceleration and jerk (per robot joint per step).

4) Benefits of Integrating Predictions: To handle the planning in dynamic workspaces, we simulate a collaborative disassembly scenario and utilize the RRT* planner to generate a set of 12 000 collision-free motions for the manipulator in one workspace that involved the current and future human motions. Figs. 7 and 8 illustrate the experimental tests based on the human motion A and B, respectively. Three cases are considered in the experimental tests: 1) planning without human arm, 2) planning without human prediction, and 3) planning with human prediction. The first case is the planning without taking into account the current position of the human arm, resulting in direct contact between the manipulator and the human. Such a case highlights the necessity of the real-time replanning in HRC scenarios. In the second case, the planning considers the current position of the human arm. When the arm is reaching and grabbing components shown in Fig. 7 (III) and Fig. 8 (III), the neural planner is capable of promptly replanning safe motions to avoid collisions. The third case is the planning with uncertainty-aware prediction. Multiple future arm poses are used to check collisions continuously. The manipulator plans motions at the early stage of the task execution since it detects potential collisions according to the predictions. In general, experimental tests show that the robot exhibits abrupt changes in motion when only considering the current human arm positions. On the other hand, by incorporating future human arm poses into the planning process, the robot demonstrates smoother motions in terms of an earlier response and a smoother path for the end-effector.

Quantitative results of smoothness evaluation in terms of velocity profile are provided in Table III. We calculate the acceleration and jerk, and take averages for each step and each robot joint. Without considering the human operator, the robot planner can find a smooth trajectory in the static environment; however, it can lead to collisions since the planner is not human-aware. When considering the human in the environment during the planning process, robot motion will be affected due to the movements of the human operator. However, the smoothness can be improved by incorporating human motion prediction in the planning process, compared with the model without human motion prediction.

VI. CONCLUSION AND FUTURE WORK

This article presented a graph-based framework that seamlessly incorporates uncertainty-aware human motion prediction into robotic motion planning. The human motion is predicted using an RNN-based prediction model, and the uncertainty of the prediction model is explicitly quantified using MCDS. The uncertainty-aware prediction is effectively integrated into a graph that represents the collaboration workspace. The manipulator motions are planned based on the constructed graph, and the uncertainty-aware prediction is utilized to expand the safety margin during the planning. The results of the experiments demonstrate that the proposed planning framework can enhance the smoothness and safety of collaborative disassembly processes.

To further enhance the safety of collaborations, future studies will focus on establishing a target inference time during the collaboration. The inference time can be determined through iterative task execution trials, where the human worker gradually increases the moving speed until contact occurs. In addition, given our heuristic approach of setting an extra radius as a safety distance, establishing the minimum safety distance can also be achieved by targeting this inference time. In addition, our forthcoming studies will involve reconstructing feasible arm poses at each predicted time instance, empirically collecting uncertainties, and presenting the results in a statistically rigorous manner to better demonstrate the validity of our approach.

REFERENCES

[1] M.-L. Lee, W. Liu, S. Behdad, X. Liang, and M. Zheng, "Robot-assisted disassembly sequence planning with real-time human motion prediction," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 53, no. 1, pp. 438–450, Jan. 2023.

[2] M.-L. Lee, X. Liang, B. Hu, G. Onel, S. Behdad, and M. Zheng, "A review of prospects and opportunities in disassembly with human–robot collaboration," *J. Manuf. Sci. Eng.*, vol. 146, no. 2, Feb. 2024, Art. no. 020902.

[3] H. Liu and L. Wang, "Human motion prediction for human-robot collaboration," *J. Manuf. Syst.*, vol. 44, pp. 287–294, 2017.

[4] D. Fridovich-Keil et al., "Confidence-aware motion prediction for real-time collision avoidance1," *Int. J. Robot. Res.*, vol. 39, no. 2/3, pp. 250–265, 2020.

[5] S. Tian, M. Zheng, and X. Liang, "TransFusion: A practical and effective transformer-based diffusion model for 3D human motion prediction," *IEEE Robot. Autom. Lett.*, vol. 9, no. 7, pp. 6232–6239, Jul. 2024.

[6] K. A. Eltouny, W. Liu, S. Tian, M. Zheng, and X. Liang, "Detgn: Uncertainty-aware human motion forecasting using deep ensembles," *IEEE Robot. Autom. Lett.*, vol. 9, no. 3, pp. 2192–2199, Mar. 2024.

[7] V. V. Unhelkar et al., "Human-aware robotic assistant for collaborative assembly: Integrating human motion prediction with planning in time," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2394–2401, Jul. 2018.

[8] H. Moudoud, Z. Mlika, L. Khoukhi, and S. Cherkaoui, "Detection and prediction of FDI attacks in IoT systems via hidden Markov model," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 2978–2990, Sep./Oct. 2022.

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

[10] W. Liu, X. Liang, and M. Zheng, "Dynamic model informed human motion prediction based on unscented Kalman filter," *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 6, pp. 5287–5295, Dec. 2022.

[11] M. Li, S. Chen, Y. Zhao, Y. Zhang, Y. Wang, and Q. Tian, "Dynamic multiscale graph neural networks for 3 d skeleton based human motion prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 214–223.

[12] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Social-Stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 14424–14432.

[13] E. Aksan, M. Kaufmann, P. Cao, and O. Hilliges, "A spatio-temporal transformer for 3D human motion prediction," in *Proc. IEEE Int. Conf. 3D Vis.*, 2021, pp. 565–574.

[14] J. Wang, H. Xu, M. Narasimhan, and X. Wang, "Multi-person 3D motion prediction with multi-range transformers," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 6036–6049, 2021.

[15] M. Faroni, M. Beschi, and N. Pedrocchi, "Safety-aware time-optimal motion planning with uncertain human state estimation," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 12219–12226, Oct. 2022.
[16] A. Kanazawa, J. Kinugawa, and K. Kosuge, "Adaptive motion planning for a collaborative robot based on prediction uncertainty to enhance human safety and work efficiency," *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 817–832, Aug. 2019.
[17] Y. Cheng, W. Zhao, C. Liu, and M. Tomizuka, "Human motion prediction using semi-adaptable neural networks," in *Proc. IEEE Amer. Control Conf.*, 2019, pp. 4884–4890.
[18] J. Zhang et al., "UC-Net: Uncertainty inspired RGB-D saliency detection via conditional variational autoencoders," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8582–8591.
[19] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, Jul. 1968.
[20] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Res. Rep. 9811, 1998.
[21] J. D. Gammell, S.S. Srinivasa, and T. D. Barfoot, "Batch informed trees (BIT*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 3067–3074.
[22] L. Janson, E. Schmerling, A. Clark, and M. Pavone, "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions," *Int. J. Robot. Res.*, vol. 34, no. 7, pp. 883–921, 2015.
[23] T. Marcucci, M. Petersen, D. von Wrangel, and R. Tedrake, "Motion planning around obstacles with convex optimization," *Sci. Robot.*, vol. 8, no. 84, 2023, Art. no. eadf7843.
[24] S. Zimmermann, G. Hakimifard, M. Zamora, R. Poranne, and S. Coros, "A multi-level optimization framework for simultaneous grasping and motion planning," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 2966–2972, Apr. 2020.
[25] L. Li, Y. Miao, A. H. Qureshi, and M. C. Yip, "MPC-MPNet: Model-predictive motion planning networks for fast, near-optimal planning under kinodynamic constraints," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4496–4503, Jul. 2021.
[26] M. J. Bency, A. H. Qureshi, and M. C. Yip, "Neural path planning: Fixed time, near-optimal path generation via oracle imitation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 3965–3972.
[27] A. H. Qureshi, Y. Miao, A. Simeonov, and M. C. Yip, "Motion planning networks: Bridging the gap between learning-based and classical motion planners," *IEEE Trans. Robot.*, vol. 37, no. 1, pp. 48–66, Feb. 2021.
[28] P. Cai, H. Wang, H. Huang, Y. Liu, and M. Liu, "Vision-based autonomous car racing using deep imitative reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7262–7269, Oct. 2021.
[29] L. Gao et al., "Cola-HRL: Continuous-lattice hierarchical reinforcement learning for autonomous driving," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 13143–13150.
[30] W. Liu, K. Eltouny, S. Tian, X. Liang, and M. Zheng, "Kg-planner: Knowledge-informed graph neural planning for collaborative manipulators," 2024, *arXiv:2405.07962*.
[31] Y. Cheng, L. Sun, C. Liu, and M. Tomizuka, "Towards efficient human-robot collaboration with robust plan recognition and trajectory prediction," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2602–2609, Apr. 2020.
[32] P. Kratzer, M. Toussaint, and J. Mainprice, "Prediction of human full-body movements with motion optimization and recurrent neural networks," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 1792–1798.
[33] P. Zheng, P.-B. Wieber, J. Baber, and O. Aycard, "Human arm motion prediction for collision avoidance in a shared workspace," *Sensors*, vol. 22, no. 18, 2022, Art. no. 6951.
[34] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn*, 2016, pp. 1050–1059.
[35] Y. Gal and Z. Ghahramani, "Bayesian convolutional neural networks with bernoulli approximate variational inference," 2015, *arXiv:1506.02158*.
[36] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.
[37] J. A. Starek, J. V. Gomez, E. Schmerling, L. Janson, L. Moreno, and M. Pavone, "An asymptotically-optimal sampling-based algorithm for bi-directional motion planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 2072–2078.

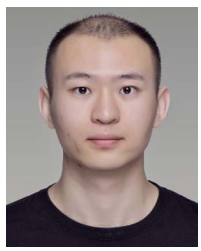
Wansong Liu received the B.S. degree in materials processing and controlling engineering from the China University of Mining and Technology, Xuzhou, China, in 2017, and the M.S. degree in 2020 in mechanical and aerospace engineering from the University at Buffalo, Buffalo, NY, USA, where he is currently working toward the Ph.D. degree in mechanical engineering.

His research interests include planning, learning, and the control of robotic manipulators in collaboration with human.

Kareem Eltouny received the B.S. degree in civil engineering from Suez Canal University, Ismailia, Egypt, in 2014, and the M.S. degree in civil engineering in 2019 from the the University at Buffalo, Buffalo, NY, USA, where he is currently working toward the Ph.D. degree in civil engineering.

He has 3 years of experience in engineering consulting firms. His research interest includes data-driven structural health monitoring, as well as developing machine learning applications for human–robot collaboration.

Mr. Eltouny was the recipient of the Fulbright Scholarship for his M.S. degree.

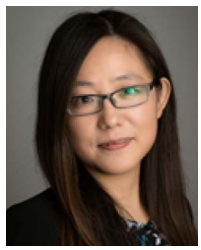


Sibo Tian received the B.S. degree in flight vehicle design and engineering from the Dalian University of Technology, Dalian, China, in 2019, and the M.S. degree in mechanical engineering from University of California San Diego, San Diego, CA, USA, in 2021. He is currently working toward the Ph.D. degree in mechanical engineering from Texas A&M University, College Station, TX, USA.

His research interests include prediction, planning, learning and control with applications to human–robot collaboration.

Xiao Liang received the B.S. degree in civil engineering from Hunan University, Changsha, China, in 2010, and the M.S. and Ph.D. degrees in civil engineering from the University of California at Berkeley, Berkeley, CA, USA, in 2011 and 2016, respectively.

He is currently an Assistant Professor with the Zachry Department of Civil and Environmental Engineering, Texas A&M University, College Station, TX, USA. Before that, he was an Assistant Professor with the Department of Civil, Structural and Environmental Engineering, University at Buffalo, Buffalo, NY, USA. His research interests include health monitoring and autonomous inspection of infrastructure systems through advanced data analytics, model-based, and machine learning.



Minghui Zheng received the B.S. degree in engineering mechanics and the M.S. degree in control science and engineering from Beihang University, Beijing, China, in 2008 and 2011, respectively, and the Ph.D. degree in mechanical engineering from University of California, Berkeley, Berkeley, CA, USA, in 2017.

She is currently an Associate Professor of the J. Mike Walker '66 Department of Mechanical Engineering, Texas A&M University, College Station, TX, USA. Before that, she was an Associate Professorwith the University at Buffalo, Buffalo, NY, USA. Her research interest lies in learning, planning, and control with applications to several areas that are of vital importance to manufacturing and robotics.

Dr. Zheng was the recipient of the NSF CAREER Award in 2021.