2020 American Control Conference
Denver, CO, USA, July 1-3, 2020

Experimental Evaluation of Human Motion Prediction Toward Safe and Efficient Human Robot Collaboration

Weiye Zhao, Liting Sun, Changliu Liu and Masayoshi Tomizuka

Abstract— **Human motion prediction is non-trivial in modern industrial settings. Accurate prediction of human motion can not only improve efficiency in human-robot collaboration, but also enhance human safety in close proximity to robots. Although many prediction models have been proposed with various parameterization and identification approaches, some fundamental questions remain unclear: what is the necessary parameterization of a prediction model? Is online adaptation of models necessary? Can a prediction model help improve safety and efficiency during human-robot collaboration? These unaddressed questions result from the difficulty of quantitatively evaluating different prediction models in a closed-loop fashion in real human-robot interaction. This paper develops a method to evaluate the closed-loop performance of different prediction models. In particular, we compare models with different parameterizations and models with or without online parameter adaptation. Extensive experiments were conducted on a human-robot collaboration platform. The experimental results demonstrate that human motion prediction significantly enhance the collaboration efficiency and human safety. Adaptable prediction models that are parameterized by neural networks achieve better performance.**

I. INTRODUCTION

Human-robot collaboration (HRC) has drawn increasing attentions in many fields due to its benefits in significantly boosting the team efficiency and flexibility. In HRC tasks such as electronics assembly [1], human workers work in close proximity to robots. To enable safe and efficient HRC, robots should be aware of current and future human movements, so as to quickly adapt their behavior to safely and efficiently collaborate with human workers [2], [3]. Human motion, however, is naturally highly nonlinear, stochastic [4], and time-varying with significant individual differences. Such characteristics makes it hard to predict. A prediction model that works for one person may not be applicable to another, or even the same person at a different time.

A prediction model can take multiple forms such as linear regression model, supported vector machine, Gaussian mixture model, hidden Markov model [5], feed-forward neural network and recurrent neural networks (RNNs) [6]. The parameters of a prediction model can either be fixed with offline training or online adapted.

This work was supported by National Science Foundation (Award #1734109)

W. Zhao and C. Liu are with the Robotic Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: weiyezha, cliu6@andrew.cmu.edu).

L. Sun and M. Tomizuka are with the Department of Mechanical Engineering, University of California, Berkeley, CA 94720 USA (e-mail: litingsun, tomizuka@berkeley.edu).

In terms of performance evaluation of a prediction model, the majority of existing works focus on quantifying its prediction accuracy on specific datasets. Few of them evaluate the effectiveness of the prediction model in real world experiments with human-robot systems. Considering the gap between the activities in the datasets and real human motion in industrial settings, it is hard to determine whether a prediction model would lead to safe and efficient human-robot collaboration if real world evaluation is missing.

This paper introduces a series of human-in-the-loop co-robot experiments to compare different prediction models. The co-robot platform is based on the safe and efficient robot collaborative system (SERoCS) [7] as shown in Fig. 1. We aim to investigate the following problems:

1) whether a complex parameterization (e.g., using a neural network) of a prediction model is necessary;
2) whether online adaptation of a prediction model is necessary;
3) whether active prediction improves safety and efficiency of human-robot collaboration.

To answer these questions, we compare four types of prediction models:

1) a linear regression model without adaptation,
2) a linear regression model with adaptation,
3) a neural network model without adaptation,
4) a neural network model with adaptation, called a semi-adaptable neural network model [2].

The metrics for performance evaluation include 1) safety and 2) efficiency of the HRC team. In the experiments, the baseline is the performance of the HRC team without active prediction, namely, when the robot only considers human motion uncertainties according to its physical constraints. The experimental results demonstrate that with active prediction, the safety score is doubled. Among the four evaluated models, the semi-adaptable neural network model achieves higher efficiency score than others, which means that prediction model with complex parameterization online adaptation can help achieve more accurate human motion prediciton.

The remainder of the paper is organized as follows. Section II formulates the human motion prediction problem and presents the four motion prediction models. Section III proposes three hypotheses about the effects of motion prediction on human-robot collaboration. Section IV describes the experimental setup, and Section V demonstrates the performance of the four models compared to the baseline. Section VI discusses the experimental results and Section VII concludes the paper.

978-1-5386-8266-1/$31.00 ©2020 AACC

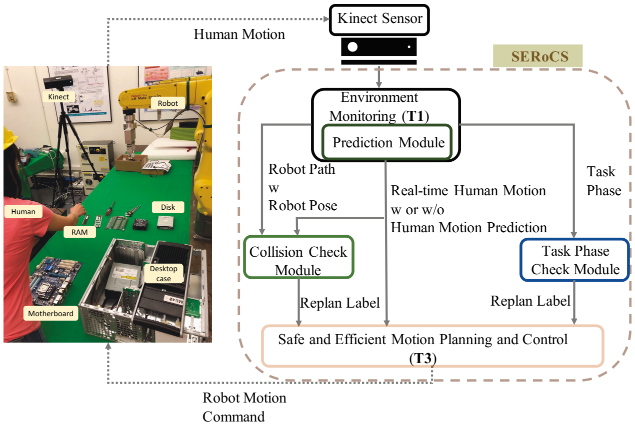

Fig. 1. The simplified safe and efficient robot collaboration system (SERoCS).

II. Human Motion Prediction

Let $x(k) \in \mathbb{R}^3$ represent the three-dimensional position of a joint in Cartesian space at time k. Define $\mathbf{x}_{+M}(k+1) \in \mathbb{R}^{3M}$ as a M-step joint positions starting at time $k+1$ and $\mathbf{x}_{-N}(k) \in \mathbb{R}^{3N}$ be a N-step joint positions before time k. Following the formulation in [2], the transition model of human motion on a selected joint can be defined as:

$$\mathbf{x}_{+M}(k+1) = f(\mathbf{x}_{-N}(k), a) + w_k, \tag{1}$$

where $a \in \mathbb{N}^1$ is a discrete action label representing different motion categories, and $w_k \in \mathbb{R}^{3M}$ is a zero-mean white Gaussian noise. The function $f(\mathbf{x}_{-N}(k), a) : \mathbb{R}^{3N} \times \mathbb{N}^1 \to \mathbb{R}^{3M}$ encodes the transition of the human motion, which takes the historical joint trajectory and current action label as inputs, and outputs the future positions of the joint.

The following subsections presents the four human motion prediction models that we evaluate in the work.

A. Linear Regression Model without Adaptation (M1)

In this model, the dynamics (1) is parameterized by a linear regression [8]:

$$f(\mathbf{x}_{-N}(k), a) = V^T s_k, \tag{2}$$

where $s_k = [\mathbf{x}_{-N}(k)^T, a, 1]^T \in \mathbb{R}^{3N+2}$ is the input vector, in which the element 1 is for the bias term, and $V \in \mathbb{R}^{(3N+2)\times 3M}$ represents the unknown model parameters to be identified from human motion dataset.

Linear regression is usually solved using the least square method [9], which is inefficient for large scale matrix inverse. To speed up the computation, we adopt stochastic gradient descent (SGD) to perform the multi-variate linear regression [10].

B. Neural Network Model without Adaptation (M2)

Feedforward Neural Networks (FNNs) has also been widely utilized to approximate the motion transition model f in (1) [11]. Parameterization using more complex models such as Recurrent Neural Networks (RNNs) will not be covered in this paper, but has been discussed in [12], [13].

In particular, we define a n-layer FNN as:

$$f(\mathbf{x}_{-N}(k), a) = W^T \max(0, g(U, s_k)) + \varepsilon(s_k), \tag{3}$$

where g denotes the first $n-1$ layers of the FNN, whose weights are packed in U. $\varepsilon(s_k) \in \mathbb{R}^{3M}$ is the function reconstruction error, which is small when the neural network is fully trained. $W \in \mathbb{R}^{n_h \times 3M}$ is the last layer parameter weights, where $n_h \in \mathbb{N}$ is the number of neurons in the hidden layer of the neural network [14]. We also deploy stochastic gradient descent to train FNN model.

C. Linear Regression Model with Adaptation (M3)

As we discussed above, human motion is time-varying and differs among individuals. Hence, it is hard to fully capture it via a fixed linear regression model as (2). To address the problem, linear regression model with parameter adaptation is proposed based on recursive least square parameter adaptation algorithm (RLS-PAA) [15], [16].

By stacking all the column vectors of V, we get a time varying vector $\theta_k \in \mathbb{R}^{3M(3N+2)}$ to represent unknown linear model parameters. We further define data matrix $\Phi_k \in \mathbb{R}^{3M\times 3M(3N+2)}$ as a diagonal concatenation of $3M$ pieces of s_k^T. With Φ_k and θ_k, the transition model (1) using linear regression model (2) can be rewritten as

$$\mathbf{x}_{+M}(k+1) = \Phi_k \theta_k + w_k, \tag{4}$$

where w_k denotes the noise at time step k. Let $\hat{\theta}_k$ denotes the parameter estimate at time step k, and let $\tilde{\theta}_k = \theta_k - \hat{\theta}_k$ be the parameter estimation error. We define the *a priori* estimate of the state and the estimation error as:

$$\hat{\mathbf{x}}(k+1|k) = \Phi_k \hat{\theta}_k, \tag{5}$$

$$\tilde{\mathbf{x}}(k+1|k) = \Phi_k \tilde{\theta}_k + w_k. \tag{6}$$

The main steps of RLS-PAA are to iteratively update the parameter estimate $\hat{\theta}_k$ and predict $\mathbf{x}_{+M}(k+1)$ when new measurements become available. The parameter update rule of RLS-PAA can be summarized as [17]:

$$\hat{\theta}_{k+1} = \hat{\theta}_k + F_k \Phi_k^T \tilde{\mathbf{x}}(k+1|k), \tag{7}$$

where F_k is the learning gain updated by:

$$F_{k+1} = \frac{1}{\lambda_1(k)}[F_k - \lambda_2(k)\frac{F_k \Phi_k \Phi_k^T F_k}{\lambda_1(k) + \lambda_2(k)\Phi_k^T F_k \Phi_k}], \tag{8}$$

where $0 < \lambda_1(k) \le 1$ and $0 \le \lambda_2(k) < 2$. Typical choices for $\lambda_1(k)$ and $\lambda_2(k)$ are:

1) $\lambda_1(k) = 1$ and $\lambda_2(k) = 1$ for typical least squares gain.
2) $0 < \lambda_1(k) < 1$ and $\lambda_2(k) = 1$ for least squares gain with forgetting factor.
3) $\lambda_1(k) = 1$ and $\lambda_2(k) = 0$ for constant adaptation gain.

D. Neural Network Model with Adaptation (M4)

In this model, the last-layer weight W in the feed-forward neural network (3) is also adaptable online using RLS-PAA [2]. We call the new model the semi-adaptable neural network [2]. This method requires the neural network to be pre-trained offline so that effective feature can be extracted when the last layer is removed [18]. To accommodate time-varying behaviors and individual differences in human motion, we just need to adjust the weights W in the last layer of the neural network.

To apply RLS-PAA on the adaptation of W in (3), we follow similar operations in M3, and transform W and $\max(0, g(U, s_k))^T$ into θ_k and Φ_k, respectively. Then (1) and (3) can be written into the same form as (4). Note that the adaptation of the last layer of a neural network doee not require time-consuming operations such as gradient computation, thus its computation cost is fairly low and is suitable for real-time deployment. The adaptation procedure is summarized in Algorithm 1.

Algorithm 1: Semi-adaptable neural network for human motion prediction

Input : Offline trained neural network (3) with g, U and W
Output : future trajectory $\mathbf{x}_{+M}(k+1)$
Variables : Adaptation gain F, neural network last layer parameters θ, variance of zero-mean white Gaussian noise $Var(w_k)$
Initialization: $F = 1000I$, $\theta =$ column stack of W, $\lambda_1 = 0.998$, $\lambda_2 = 1$

while *True* **do**
    Wait for a new joint position p captured by Kinect and current action label a from action recognition module;
    Construct $s_k = [\mathbf{x}_{-N}(k), a, 1]^T$;
    Obtain $\Phi(k)$ by diagonal concatenation of $max(0, g(U, s_k))$;
    Update F by (8);
    Adapt the parameters θ in last layer of neural network by (7);
    Calculate future joint trajectory $\mathbf{x}_{+M}(k+1)$ by (4);
    send $\mathbf{x}_{+M}(k+1)$ to robot control.
end

III. HYPOTHESIS

Based on the prediction models discussed in the previous section, we anticipate that active prediction will affect both safety and efficiency during human-robot collaboration, and the effects vary for models with different parameterizations and models with or without online adaptation. Here we propose three main hypotheses, which will be verified in the experiments to be discussed in the following sections:

Hypothesis 1 (Prediction Accuracy) *Online adaptation of a prediction model improves the prediction accuracy. The prediction accuracy is higher for models that can encode nonlinear features.*

Hypothesis 2 (Prediction and Safety) *Active human motion prediction enables the robot motion planner to take human tendency into consideration, which improves human safety.*

Hypothesis 3 (Prediction and Efficiency) *Collaboration efficiency is higher if the prediction models can achieve higher prediction accuracy (e.g., adaptable prediction models).*

IV. EXPERIMENT DESIGN

To quantitatively evaluate the effects of motion prediction on human-robot collaboration, we conduct a series of experiments in which a human works in close proximity to a robot while the robot is performing predefined tasks.

A. Experiment Setup

The experiment platform is shown in the left part of Fig. 1. The robot manipulator is FANUC LR Mate 200iD/7L, a 6-degree-of-freedom industrial robot. There is one Kinect sensor to monitor the environment. We track the trajectory of the human's right wrist. All the experiments are implemented in Matlab 2016 platform on a Windows desktop with 2.7 GHz Intel Core i5 Processor and 16 GB RAM. The robot controller runs on the Simulink RealTime target.

B. The Safe and Efficient Robot Collaboration System

The experiment is built upon the safe and efficient robot collaboration system (SERoCS) [7]. SERoCS consists of three modules: (T1) the robust cognition module for environment monitoring and prediction, (T2) the optimal task planning module for efficient human-robot collaboration, and (T3) the motion planning and control module for safe human-robot interaction. In the experiments, the robot is only required to track a simple trajectory, which does not require task planning in T2. We close the loop with only T1 and T3. The input to T1 is the real-time measurement from the Kinect sensor, while the output from T1 is the predicted trajectory $\hat{\mathbf{x}}(k+1 \mid k)$. All four prediction models described in Section II are used separately in T1 for comparison.

We also deploy the T3 module to plan and control the robot motion. The input to T3 is the environment information and the predicted human motion from T1, while the output is the planned robot motion trajectory $\mathbf{x}_R$. The robot is equipped with 1) a short-term safety-oriented planner based on the safe set algorithm [15] and 2) a long-term efficiency-oriented planner based on the convex feasible set algorithm [19]. The two planners run in parallel.

The closed-loop execution of the human-robot collaboration system is summarized below, which is also shown in Fig. 1.

1) The T1 module estimates the historical human states $\mathbf{x}_{-N}(k)$ from the sensory data and predicts the future trajectory $\hat{\mathbf{x}}(k+1 \mid k)$.
2) The T3 module plans the future robot trajectory given information from the T1 module. The short term planner runs in receding horizon for 1 kHz. The long term planner only replans in two scenarios: when a new task is specified, or when the safety specification is triggered, i.e., the distance between the human and the robot is below a threshold.
3) The planned trajectory is send to the robot hardware for execution. New states of the system will be obtained in the next time step, and the steps 1) -3) repeat until experiments end.

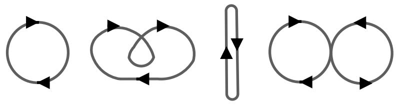

Fig. 2. Four predefined motions for human right hand.

C. Human-Robot Collaboration Tasks

To effectively compare the effects of the four motion prediction models in Section II, we need to design human-robot interactive tasks considering the following features.

1) Diversity of human motions: The improvement brought by motion prediction should be robust to diverse complex human motions. Therefore, the task needs to emphasize the diversity of human motions.

2) Responsiveness of the robot: We want to test if the robot can operate safely and efficiently in the worst case scenario, i.e., when the human completely ignores the robot. In this situation, the robot needs to react quickly to meet the human's needs.

3) Repeatability: All experiments should be repeatable. Robot should be required to perform the same task at each trial. The same motion planner should be applied. Different human motion types should also be designed and fixed across comparison experiments with different motion prediction methods.

To meet these requirements, we design the following interactive tasks, which are commonly encountered in workspace-sharing human-robot interactions in factory settings. The situation is that the human subject is working on his/her own task and the robot wants to fetch a target object to help the human. As shown in Fig. 1, we allocate a Disk and a RAM for the robot to fetch, starting from its idle pose. To emphasize the diversity of human motions, we design four different motion patterns, as shown in Fig. 2 and ask the human subjects to follow. The four motion patterns represent four different action labels a in (1). The action label is pre-defined across experiments. To test the responsiveness of the robot, we instruct the human subject in the experiments not to respond to the robot as long as safety is under control with the emergency brake. To guarantee the repeatability, we fix the locations of the Disk and RAM across different experiments. This paper considers one human subject across all experiments, which is enough to verify the effectiveness of different prediction methods, since the trajectories from one human subject are already diverse.

D. Experiment Procedure

1) Data collection: For each type of motion defined in Fig. 2, we collect 30 trajectories from the human subject, and use the trajectories to train the four prediction models. Each trajectory contains 45 time steps, where the human subject conducts routine motions without the robot. The sampling rate is 5 Hz. A low-pass filter is applied to smooth the trajectories before training. We set $N = M = 3$ throughout the experiments.

2) Offline training: We pre-train the prediction models using the collected data. For M1, we use a 10×9 matrix to represent transformation parameter θ, and apply SGD to optimize θ. For M2, we apply a 3-layer fully-connected neural network with a structure of $11 \times 40 \times 9$. The loss function is set to be the L2 loss. All the learning rates are set to 0.001 and the number of epochs is 100. Note that the parameters of M3 and M4 can be initialized with the parameters of the pre-trained M1 and M2, respectively.

3) Experimental validation: After the off training, we start human-robot interaction experiments by substituting the prediction module with the four different prediction models. For each human motion shown in Fig. 2, 20 independent trials are conducted with each prediction model. In each trial, 45 sampling points are recorded which contain the human trajectory $\bar{\mathbf{x}}$, the robot trajectory, and human motion prediction $\hat{\mathbf{x}}$ at each time steps. We also record the same amount of baseline trials, namely, the joint motions of the human and the robot without active prediction.

E. Evaluation Metrics

1) Prediction Accuracy: We define the average prediction error using average distance between the predicted trajectory and ground truth trajectory.

$$\text{Prediction Error} = \frac{1}{n}\sum_{i=1}^{n} \|\bar{\mathbf{x}}(i+1) - \hat{\mathbf{x}}(i+1)\|_2, \qquad (9)$$

where a smaller average prediction error implies better prediction performance.

2) Safety: The robot is supposed to keep a proper distance from the human to avoid potential collisions. Existing works measure safety during HRI by the distance between human and robot [20]. Similarly, we define the safety index for each trail as the average closest distance between the human and the robot:

$$\text{Safety} = \frac{1}{n}\sum_{i=1}^{n}(\min(Dist(H_i, R_i))), \qquad (10)$$

where n is the sample frame number for a trial, H_i and R_i denote human pose and robot pose at frame i, respectively. A higher safety index means the robot is farther from the human, hence safer.

3) Efficiency: For human-lead HRI scenarios where the human has priority to occupy the workspace, the robot should wait until human leaves before it can approach its target, whereas human's work time is independent of robot's efficiency. Therefore, conventional efficiency metrics by evaluating how many target can be achieved by the robot within certain amount of time [21] does not fit in those scenarios and better efficiency metrics should be developed. Note that proper prediction of human motion can make the robot escape in advance when human is approaching, and continue with its task when the human tends to get away. In other words, good motion prediction can improve the efficiency of robot motion by making the robot keep as close to its target as possible. Denote the average robot-target distance without human interference as ground truth $\mathbb{D}_{RT}$. We define the efficiency index by comparing the average distance

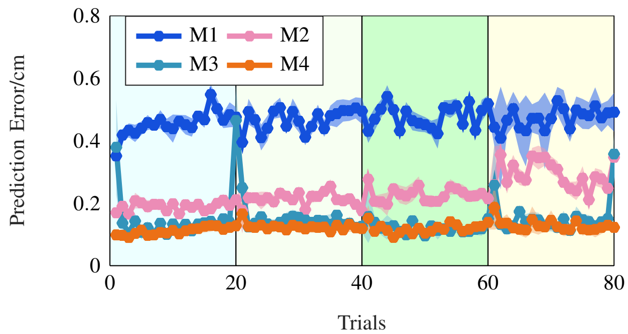

Fig. 3. Prediction error comparison among four different motion prediction algorithms on 80 trials. The bold lines are averaged over all sample points for each trial, the shaded area presents its standard deviation. Rectangle backgrounds with different color denote the different motion classes.

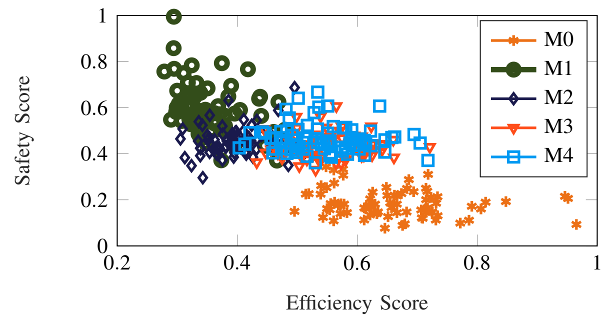

Fig. 4. Safety and efficiency score map for four motion prediction algorithms and collaboration system without prediction. M0 denotes result without prediction.

TABLE I

MEAN AND VARIANCE OF THE PREDICTION ERROR FOR FOUR MOTION PREDICTION MODELS.

	prediction error mean (m)	prediction error variance (m^2)
M3	0.1447	0.0032
M4	**0.1209**	**0.0002**
M2	0.2304	0.0021
M1	0.4678	0.0012

TABLE II

MEAN AND VARIANCE OF THE SAFETY AND EFFICIENCY SCORES FOR THE FOUR MOTION PREDICTION MODELS.

	safety mean	safety variance	efficiency mean	efficiency variance
M3	0.5577	0.2593	0.5741	0.0162
M4	**0.5603**	**0.1353**	**0.5466**	**0.0055**
M2	0.5486	0.2775	0.3983	0.0026
M1	0.9209	1.130	0.3541	0.0024
w/o Prediction	0.2550	0.1898	0.6588	0.0095

between the robot and its target with the ground truth $\mathbb{D}_{RT}$:

$$\text{Efficiency} = \frac{\mathbb{D}_{RT}}{\frac{1}{n}\sum_{i=1}^{n}(Dist(R_i,T))}, \tag{11}$$

where T denotes the target position. A higher efficiency index indicates that the robot completes tasks more efficiently.

V. RESULTS

A. Prediction accuracy

We first evaluate the performance of the four prediction models according to the prediction error as shown in Fig. 3. Semi-adaptable neural network results in much smaller prediction error as well as standard deviation on the 80 trials. The statistics are also summarized in Table I.

B. Safety and efficiency

We also compare the safety and efficiency scores for the four prediction models and baseline (i.e., the scenario without prediction). Comparison results is shown in Fig. 4.

Compared with the scenario without prediction, active prediction using either prediction model boosts the safety scores with a significant margin. Adaptable models achieve the same safety level as non-adaptable models, while the efficiency scores of adaptable models are not compromised compared to the baseline. The mean and variance of safety-efficiency score of the five methods are shown in Table II. Semi-adaptable neural network leads to the most robust performance with the smallest variance, and its performance is also well balanced in terms of safety and efficiency.

We also applied the unpaired t-test to determine the significance of improvement of active prediction and adaptation techniques in the statistical sense. The t value in terms of safety score between the neural network group and that without prediction is 3.8413 and the corresponding two-tailed P value is 0.0002. On the other hand, the t value in terms of efficiency score between M4 and M2 is 14.7321 and the corresponding two-tailed P value < 0.0001. The above null hypothesis testings demonstrate that the improvements brought by prediction and online adaptation in terms of safety and efficiency are considered to be statistically significant.

VI. DISCUSSION

A. Hypothesis 1 - Online Adaptation Improves Accuracy

As shown in the results, the prediction error is reduced significantly for adaptable models, which supports that online adaptable of the prediction model can improve prediction performance (**Hypothesis 1**). The quantitative metrics of prediction error mean and variance indicate that the semi-adaptable neural network leads to the best performance (**Hypothesis 1**). Due to high non-linearity of human motion, models parameterized by M2 perform better than models parameterized by M1.

Encoding nonlinear features will improve the prediction performance. M3 only takes small quantities of past joint positions as input, and fails to encode the nonlinear features from input. Thus, small input noise can cause large prediction variance. However, M4 encodes enough nonlinear features for adaptation from the raw input. Large quantities of nonlinear feature make online adaptation less sensitive to noisy input. Thus we observe a small prediction error variance for M4. Note that M2 performs worse than M3; this is because the fixed offline trained model can not capture the time-varying behavior of human motion. If the human motion in testing phase differs from that in training phase greatly, a large prediction error will be observed for offline models.

B. Hypothesis 2 - Effective Prediction Improves Safety

It is notable that the average human-robot distance is only 0.255 m without prediction, which is less than the minimum threshold 0.3 m for safe human-robot interaction. However, the average safety scores are doubled when prediction modules are applied (**Hypothesis 2**). It is very common that human might approach to a robot at a high speed, which leaves the robot a shorter time to escape the potential collision, especially when robot system update frequency is low.

The prediction models predict the tendency of human motion, such that the robot can generate trajectories by taking the future constraints into consideration. When the human is moving fast toward robot, the potential collision will be detected in a timely manner. Thus replanner can be triggered in advance to maintain safe human-robot distance.

C. Hypothesis 3 - Effective Prediction Improves Efficiency

Though safety score is largely enhanced with active prediction, efficiency scores of prediction models without online adaptation are greatly compromised, since the robot's behavior is too conservative. However, when online adaptation is applied, the safety score is well maintained and the efficiency score is greatly boosted to the same level as the scenario without prediction (**Hypothesis 3**).

Good human-robot collaboration system should excel both in safety and efficiency. When human motion prediction is good, the robot will accurately capture the tendency when human is getting away. Thus, the robot can quickly resume its own task by planning a new path that bypasses the predicted human trajectory. In such scenario, human safety is guaranteed and the robot does not need to wait too long or detour too much. However, if human motion prediction is inaccurate, unrealistic path planning will be produced. As a result, efficiency is deteriorated and collision might also happen.

VII. Conclusion

In this paper, we quantitatively evaluated the effects of human motion prediction on human-robot collaboration. We designed a series of human-robot interaction experiments. We compared models with different parameterizations, and models with and without online parameter adaptation. The experiment results demonstrated that human motion prediction significantly enhanced the collaboration safety, and more accurate prediction led to better efficiency. Both complex parameterizations and online adaptation helped to improve the motion prediction performance. Adaptable prediction models that were parameterized by neural networks achieved the best and robustest performance in our experiments.

References

[1] P. A. Lasota and J. A. Shah, "Analyzing the effects of human-aware motion planning on close-proximity human–robot collaboration," *Human factors*, vol. 57, no. 1, pp. 21–33, 2015.

[2] 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.

[3] 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.

[4] Z. Peng, T. Genewein, and D. A. Braun, "Assessing randomness and complexity in human motion trajectories through analysis of symbolic sequences," *Frontiers in human neuroscience*, vol. 8, p. 168, 2014.

[5] D. Wu and L. Shao, "Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 724–731.

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

[7] C. Liu, T. Tang, H.-C. Lin, Y. Cheng, and M. Tomizuka, "Serocs: Safe and efficient robot collaborative systems for next generation intelligent industrial co-robots," *arXiv:1809.08215*, 2018.

[8] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, and H.-P. Seidel, "A statistical model of human pose and body shape," in *Computer graphics forum*, vol. 28, no. 2. Wiley Online Library, 2009, pp. 337–346.

[9] A. Sen and N. Sinha, "On-line estimation of the parameters of a multivariable system using matrix pseudo-inverse," *International Journal of Systems Science*, vol. 7, no. 4, pp. 461–471, 1976.

[10] Y. Liu, J. Sheng, and R. Ding, "Convergence of stochastic gradient estimation algorithm for multivariable arx-like systems," *Computers & Mathematics with Applications*, vol. 59, no. 8, pp. 2615–2627, 2010.

[11] M. Bataineh, T. Marler, K. Abdel-Malek, and J. Arora, "Neural network for dynamic human motion prediction," *Expert Systems with Applications*, vol. 48, pp. 26–34, 2016.

[12] A. Abuduweili, S. Li, and C. Liu, "Adaptable human intention and trajectory prediction for human-robot collaboration," in *AAAI 2019 Fall Symposium Series, AI for HRI*, 2019.

[13] A. Abuduweili and C. Liu, "Robust online model adaptation by extended kalman filter with exponential moving average and dynamic multi-epoch strategy," in *Learning for Dynamics and Control Conference*, 2020.

[14] H. C. Ravichandar and A. P. Dani, "Human intention inference using expectation-maximization algorithm with online model learning." *IEEE Trans. Automation Science and Engineering*, vol. 14, no. 2, pp. 855–868, 2017.

[15] C. Liu and M. Tomizuka, "Control in a safe set: Addressing safety in human robot interactions," in *Proceedings of the ASME Dynamic Systems and Control Conference (DSCC)*, 2014, p. V003T42A003.

[16] ——, "Safe exploration: Addressing various uncertainty levels in human robot interactions," in *Proceedings of the American Control Conference (ACC)*. IEEE, 2015, pp. 465–470.

[17] G. C. Goodwin and K. S. Sin, *Adaptive filtering prediction and control*. Courier Corporation, 2014.

[18] B. Athiwaratkun and K. Kang, "Feature representation in convolutional neural networks," *arXiv preprint arXiv:1507.02313*, 2015.

[19] C. Liu, C.-Y. Lin, and M. Tomizuka, "The convex feasible set algorithm for real time optimization in motion planning," *SIAM Journal on Control and Optimization*, vol. 56, no. 4, pp. 2712–2733, 2018.

[20] A. M. Zanchettin, N. M. Ceriani, P. Rocco, H. Ding, and B. Matthias, "Safety in human-robot collaborative manufacturing environments: Metrics and control," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 882–893, 2015.

[21] T. Wei and C. Liu, "Safe control algorithms using energy functions: A unified framework, benchmark, and new directions," in *IEEE Conference on Decision and Control*. IEEE, 2019, pp. 238–243.