Preemptive Motion Planning for Human-to-Robot Indirect Placement Handovers

Andrew Choi[1], Mohammad Khalid Jawed[2], and Jungseock Joo[3]

Abstract— As technology advances, the need for safe, efficient, and collaborative human-robot-teams has become increasingly important. One of the most fundamental collaborative tasks in any setting is the object handover. Human-to-robot handovers can take either of two approaches: (1) direct hand-to-hand or (2) indirect hand-to-placement-to-pick-up. The latter approach ensures minimal contact between the human and robot but can also result in increased idle time due to having to wait for the object to first be placed down on a surface. To minimize such idle time, the robot must preemptively predict the human intent of where the object will be placed. Furthermore, for the robot to preemptively act in any sort of productive manner, predictions and motion planning must occur in real-time. We introduce a novel prediction-planning pipeline that allows the robot to preemptively move towards the human agent's intended placement location using gaze and gestures as model inputs. In this paper, we investigate the performance and drawbacks of our early intent predictor-planner as well as the practical benefits of using such a pipeline through a human-robot case study.

I. INTRODUCTION

Industrial robots have traditionally been utilized in caged or taped-off settings for the safety of nearby human operators. These robots, often "hard-coded", are assigned highly specified and repetitive tasks requiring a large amount of precision. As technology advances, an increase in the need for more intelligent robots has been seen as robots are applied to more dynamic, human-populated settings such as warehouses, restaurants, and public streets. One of the most challenging aspects of integrating robots into everyday life is in developing systems that can safely work with humans with minimal human supervision. In order to ensure safe and fluid operation, robots must be able to infer human intent and factor in such information into their actions accordingly, much like real human workers.

One of the most fundamental tasks between workers in a team environment is the handover of tools or materials. Direct mid-air hand-to-hand transfers require the robot to move in a way that avoids collision with the human operator while making contact with the held out object. This requires accurate estimation of the transfer point [1] as well as grasp classification [2]. Indirect placement transfers (i.e. placing
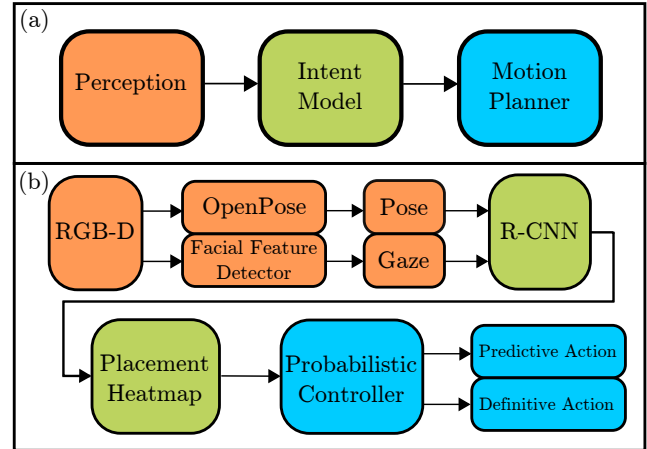

Fig. 1. Robot system pipeline. (a) shows the high level modules of the system which are color coded. (b) shows the individual components making up each of the high level modules. Predictive and definitive actions are defined in Sec. III-F.

an object down so that the opposite agent can pick it up easily) circumvent this problem entirely albeit with reduced efficiency due to the extra step of having to place the object down. Still, indirect placement transfers are heavily prevalent all throughout daily life such as when dealing with dangerous tools (passing knives) or transferring objects to occupied individuals (placing empty glasses at a bar for a bartender). Therefore, with accurate prediction of the object placement intent along with preemptive planning, robots can take advantage of the benefits of indirect handovers while minimizing inherent inefficiencies.

In this paper, we investigate the problem of preemptive motion planning for efficient human-to-robot indirect object placement transfers using real-time predictions of human object placement intent as feedback. We introduce a prediction-planning pipeline that allows the robot to preemptively move towards objects placed by a human. The primary contributions for our work are as follows: (1) we introduce a prediction-planning pipeline for preemptive human robot placement handovers that can serve as a basis for future system implementations; (2) we show practical quantitative results in a human-robot-collaboration case study utilizing preemptive motion planning when compared to more traditional wait-and-act approaches; (3) we analyze human social perception of preemptive robots; and (4) we show that accurate real-time performance can be achieved without expensive prohibitive equipment such as motion capture or gaze tracking eyewear.
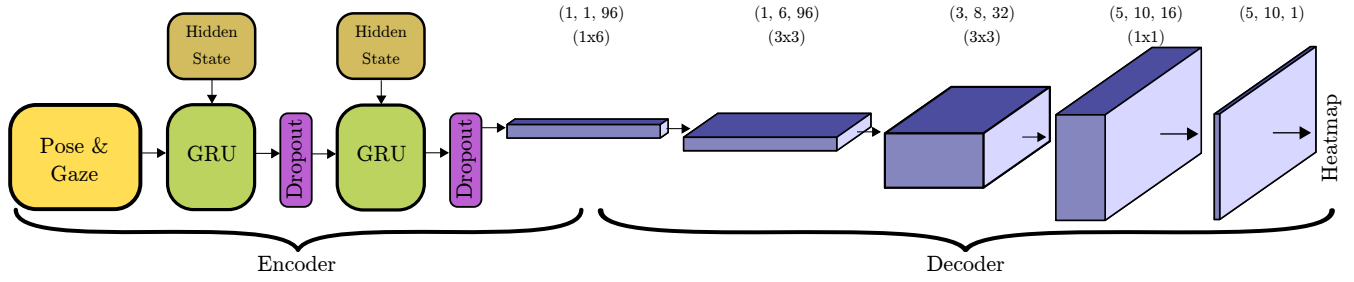
Fig. 2. An illustration of the intent model architecture. As shown, the model is a recurrent convolutional network. Recurrent units act as the encoder, where the pose & gaze feature vector is fed into the gated recurrent units (GRU) at every time step. Here, the hidden states fed back into the GRUs represent the encoded state of past time steps. Convolutional layers act as the decoder which perform a series of transpose convolutions from a 1-D encoded state vector. The dimensions of the convolutional layers are shown for a heatmap output of size 5×10, which is the size used in the case study.

The remainder of the paper is as follows. In Sec. II, we review prior work on human intent prediction and preemptive control. In Sec. III, we discuss the overall methodology involving the intent model and how predictions are factored into the robot's motions. Next, in Sec. IV, we go over the experiment design while in Sec. V, we analyze the results from the case study. Finally, conclusive remarks as well as potential future research directions are discussed in Sec. VI.

II. RELATED WORK

Human intent prediction has been a popular research topic tackled by a variety of communities in machine learning, computer vision, and robotics [3], [4]. Intent prediction has especially become a point of interest for the robotics community as humans become increasingly integrated into robot workspaces. Such inference is necessary for a variety of human-robot-collaborative tasks such as load sharing co-manipulation [5], [6], socially-aware navigation [7], [8], and object placement handovers [9], [10], [11], which commonly adopt non-verbal cues such as gesture, gaze, and facial expression to predict human intention or attention [12], [13], [14], [15], [16], [17], [18].

In terms of the social and cognitive aspects of the human-robot handover task, previous research has focused on understanding human preference of handovers [19], [20], the effect of gaze [21], [22], [23], [24], human comfort [25], as well as the roles of trust [26], [9] and support [11]. Attempts to develop frameworks that codify the coordination structure of handovers have also been made [27].

At a lower level, previous methods for human intent prediction have utilized Gaussian mixture models (GMM) to learn a library of human motions for early prediction of human reaching motions [28]. Later work showed quantitative results showcasing the benefits of utilizing GMM produced early predictions in tasks with humans and robots in the same workspace [29].

More recently, deep learning methods have also been used for intent prediction. Recurrent neural networks (RNN) have been shown to be able to classify high level human actions using gaze and body cues [23]. More similar to our specific problem, RNNs have also been shown capable of predicting object placement intents when used with Mixture Density Networks [10]. Other methods have attempted to use

inverse reinforcement learning approaches in order to obtain a cost function explaining human behavior from human demonstrations [30], [7], [31], or to learn how to interpret human intent directly from interaction and communication without demonstrations [32].

Following this, the robotics community has utilized such human intent prediction methods in order to develop preemptive control methods. Such prior work has worked on developing robot systems that are capable of preemptively planning with human intent factored in [29], [33], [34]. Similar to these works, we introduce a novel preemptive control method using data-driven intent prediction that is specifically catered towards human-to-robot object placement handovers. Furthermore, we also investigate the impact of preemptive planning on human comfort and preference.

III. METHODOLOGY

A. Intent Prediction

For the predictive model, we train a recurrent convolutional network as shown in Fig. 2. This model takes in features constructed from human gesture and gaze in order to produce a $n \times m$ heatmap representing the human agent's object placement likelihoods on an opposing flat surface. We incorporate both pose and gaze as inputs as a combination of both has been shown to produce better predictive results than one alone [23].

B. Features

For the first set of features, we simply obtain the 3D positions of the palm, elbow, and shoulder which are obtained through Openpose [35], [36]. The second set of features involve the head pose and norm. To avoid the complexities of pupil tracking, we use facial feature detection [37] to obtain the head pose $\mathbf{h}$ as well as the concurrent reference frame where the head norm is the x-axis.

For the next feature, we compute the intersection point ψ of the human's gaze vector with the table plane. To obtain this, we first tilt the head norm vector 30 degrees along the y-axis to point towards the table and treat this subsequent vector $\mathbf{g}$ as the gaze vector. Then, to obtain the gaze intersection point along the table plane, we compute

$$\psi = \mathbf{h} - \left(\frac{\mathbf{n}_t \cdot (\mathbf{h} - \mathbf{p})}{\mathbf{n}_t \cdot \mathbf{g}} \right) \mathbf{g}, \tag{1}$$
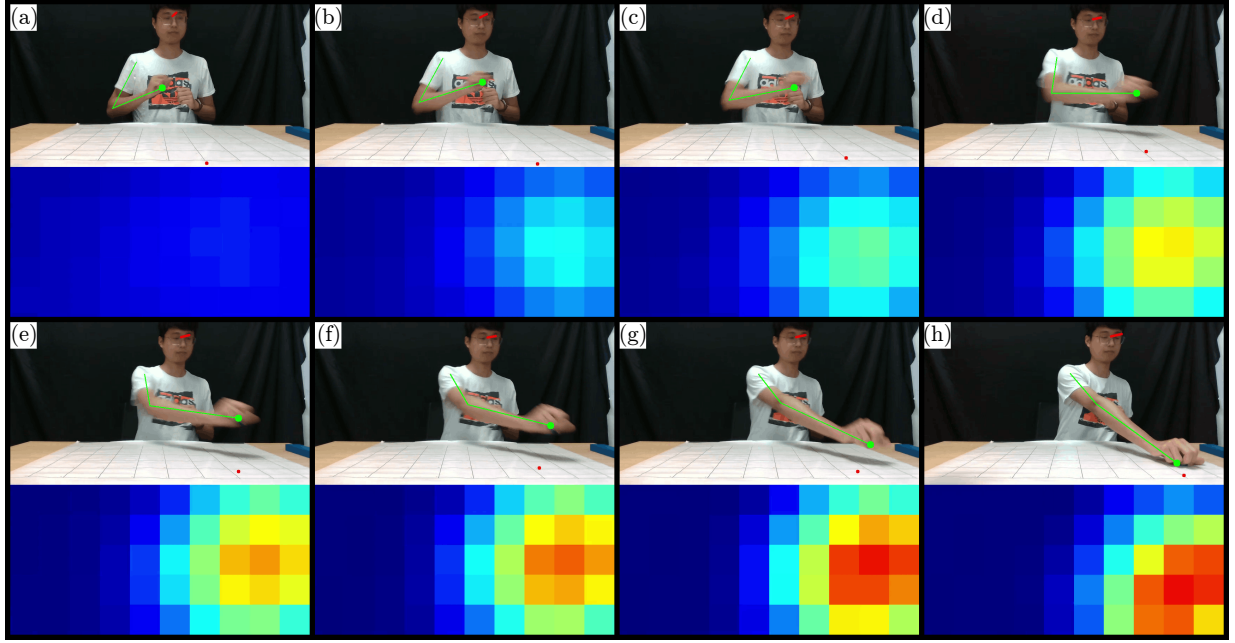
Fig. 3. An example prediction sequence. The red vector located on the participant's head is the detected face norm. The red dot indicates the gaze and table plane intersection point ψ discussed in Sec. III-B. The green skeleton shows the detected shoulder, elbow, and palm positions. For each frame (a-h) along the trajectory, the corresponding heatmap can be seen underneath. Note that at the start of the trajectory, the model is unsure of the human's placement intent. As more and more of trajectory is realized, a prediction with increasing confidence can be seen being made.

where $\mathbf{n_t}$ is the norm of the table plane and $\mathbf{p}$ is an arbitrary point on the table plane. As we normalize the table plane to be through the origin, we can remove the z-component of ψ so that ψ consists of only the x and y coordinates. These features as well as their velocities computed from the previous set of features round out the inputs to our model.

C. Labels

To train the model, we use a $n \times m$ grid of probabilities as our labels. A sigmoid layer is used as our output layer to avoid near zero probability values as the grid size increases. Furthermore, to avoid overconfident predictions resultant of one-hot encoding, we use a 2D Gaussian to employ label smoothing for a label

$$l(x,y) = \frac{1}{2\pi s_x s_y} \exp\left(-\frac{(x-m_x)^2}{2s_x^2} - \frac{(y-m_y)^2}{2s_y^2}\right), \quad (2)$$

where $x \in [0, n-1]$ and $y \in [0, m-1]$ are the label indices with respect to the grid, m_x and m_y are the true positions, and s_x and s_y are the standard deviations for the Gaussian.

Additionally, we also scale the labels by weights

$$c_t = 1 - \exp\left(-\frac{5t}{T}\right), \quad (3)$$

where t is the current time step, and T is the total length of the trajectory. This is done so that the model is initially unsure and increases its confidence as more of the trajectory is observed. To properly apply the confidence weight, we normalize the labels so that the maximum value is 1 to obtain our final label

$$l_t(x,y) = \frac{l(x,y)}{\max(\mathbf{L})} c_t, \quad (4)$$

where $\mathbf{L}$ is the set of all labels.

D. Loss

Similar to the labels, we assign the weights from Eq. 3 to our training loss as well so that earlier incorrect predictions are penalized less than later ones. As we are working with a probability regression problem, we employ mean squared error (MSE) as our loss function. The loss for one training trajectory can then be computed as

$$loss = \sum_{t=0}^{T-1} \left(\sum_{y=0}^{m-1}\sum_{x=0}^{n-1} \text{MSE}(l_t(x,y), p_t(x,y))\right) c_t, \quad (5)$$

where $p_t(x,y)$ is the prediction of the x-th and j-th grid location.

E. Weighted Predictions

Finally, when obtaining predictions during real-time operation, we keep the last h predictions in order to obtained weighted grid probabilities. Similar to [33], we use a set of weights

$$w_i = (1-\epsilon)^i \ \forall \ i \in [h], \quad (6)$$

where ϵ is a decay factor. Applying these weights allow for more recent predictions to have exponentially more influence. The weighted prediction for a certain grid location and current time step can then be computed as

$$\bar{p}(x,y) = \frac{1}{h}\sum_{i=0}^{h-1} p_i(x,y) w_i. \quad (7)$$

F. Preemptive Motion Planning

To carry out motion planning, we employ stochastic trajectory optimization for motion planning (STOMP) [38] as this algorithm has been shown to be capable of producing
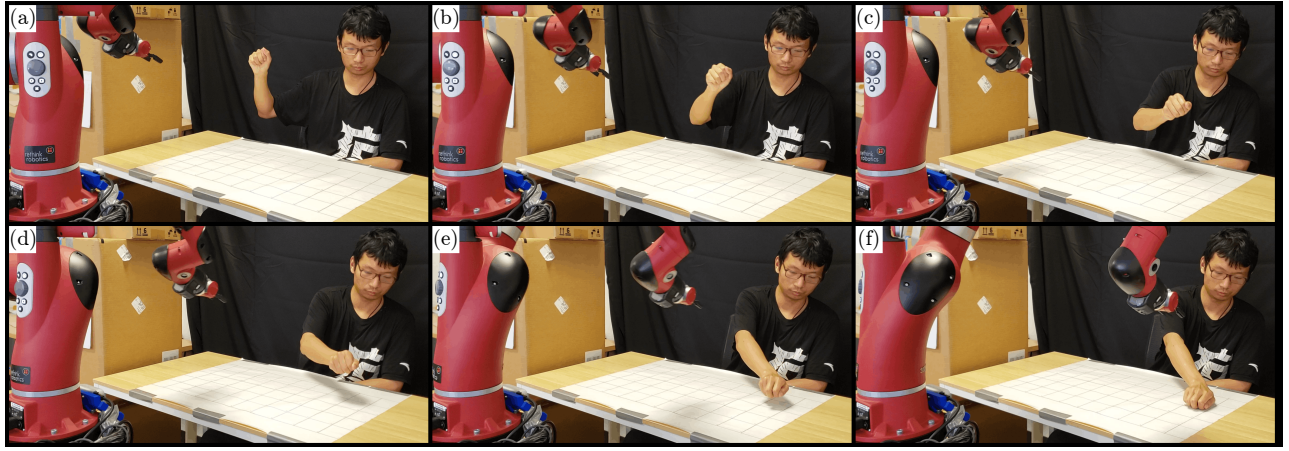
Fig. 4. Example sequence with preemptive motion planning. The robot is in the ready position in frame (a). As more and more of the trajectory is observed, the robot correctly predicts the human's placement intent and is able to move to a position to pick up the object as shown in frame (f).

smooth trajectories in real-time [28]. At every time step, we calculate the full weighted heatmap $\bar{\mathbf{p}}$ which can then be used to obtain the grid location containing the highest probability $\hat{p} = \max(\bar{\mathbf{p}})$. If $\hat{p}$ surpasses an execution limit γ and the final object location is still unknown, we have the robot carry out a *predictive action* towards the grid. Otherwise, if the final object location becomes known, we carry out a *definitive action* to pick up the object.

Two separate threads are used to asynchronously carry out predictive and definitive actions based on the current known state. If a predictive action is already in progress when the final object position becomes known, we preempt and replan a definitive action. If an ongoing predictive action is close enough (*we used a tolerance of 1 x-grid and 2 y-grids*) to the known goal position, we carry out the full motion in order to minimize any stop-and-go stuttering. The predictive and definitive algorithms are shown in Algs. 1 and 2, respectively.

IV. EXPERIMENTAL DESIGN

To analyze the benefits of using our prediction-planning pipeline, we perform a comparative case study for the human-to-robot object placement transfer task. For this task, the human places down wooden cubes measuring 1 inch on all sides on a table for the robot to pickup and sort. In an effort to minimize idle time and maximize efficiency, the robot will attempt to preemptively move to where the human places the object before the final object position becomes known.

A. Baseline

For our baseline, we create a naive planning pipeline that works purely reactively. In other words, the robot does not take into consideration any human-related factors and simply plans to grab the cube once the cube is detected. We will call this approach *reactive* and our proposed method as *preemptive*. Both methods were set to use identical joint speeds for fair comparison and consisted of three sequential motions: x-y traversal, move to pre-grasp position, and move to grasp position. More sophisticated baselines such as following the

---

Algorithm 1 Predictive Node

memory ← queue of size $h \times n \times m$ initialized to all zeros
while running and no object is detected **do**
    $\mathbf{s}$ ← obtain skeleton from Openpose
    $\mathbf{g}$ ← obtain gaze from facial feature detection
    $\mathbf{f}$ ← concatenate $\mathbf{s}$ and $\mathbf{g}$
    $\mathbf{p}$ ← forward pass with $\mathbf{f}$
    memory.push($\mathbf{p}$)
    $\bar{\mathbf{p}}$ ← obtain weighted predictions from Sec. III-E
    $\hat{p}$ ← $\max(\bar{\mathbf{p}})$
    if $\hat{p} > \gamma$ **then**
        (x, y) ← $\text{argmax}(\bar{\mathbf{p}})$ ▷ our current guess
        if no predictive action in progress **then**
            carry out *predictive action* towards (x, y)
        else if previous goal is not within tolerance **then**
            preempt and carry out new *predictive action*

Algorithm 2 Definitive Node

while running **do**
    object position ← detection algorithm
    if object is detected **then**
        if no predictive action in progress **then**
            carry out *definitive action*
        else if previous goal not within tolerance **then**
            preempt and carry out *definitive action*

---

arm trajectory were forgone due to an excessive amount of stop-and-go stuttering caused by constant motion replanning (something our preemptive approach avoids).

B. Experiment Setup

We design an experiment where human participants sequentially place wooden cubes on specified locations on a 5×10 grid where each grid is identifiable by a numeric id. Eleven grid slots were chosen at random and used as the placement sequence for the experiment. These slots can be seen listed in Fig. 5. Each grid consists of an area measuring
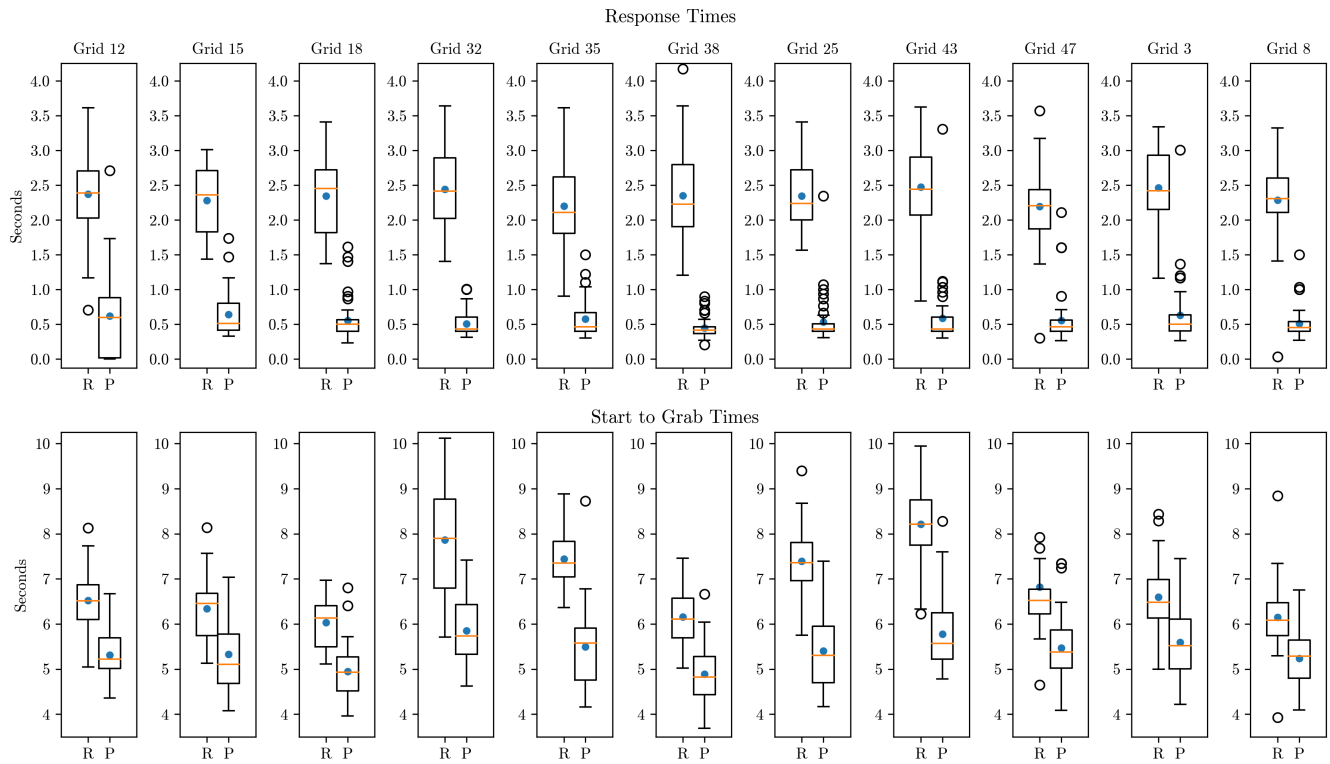
Fig. 5. The first row and second row are the response and start-to-grab time boxplots, respectively. Each column corresponds to one of the eleven grid locations used in the experiment. Each subplot consists of two boxplots labeled R for reactive and P for preemptive. The yellow line corresponds to the median while the blue dot is the mean. As shown, there is a clear decrease in both response and start-to-grab times when employing preemptive control.

8×8cm resulting in a total rectangular workspace of 0.4×1m. Prior to the experiment, we recorded and labeled a set of 76 trajectories of variable length to use as the training data.

Each participant sits in front of a table with the grid and a Rethink Robotic's Sawyer 6DOF manipulator as shown in Fig. 4. For perception, two Intel Realsense D435 cameras were used. One was mounted on the robot's base frame positioned to view and obtain human features while the second was mounted above pointing down towards the grid in order to detect the pose of the placed cubes. Fifteen participants were used in total. Each participant was informed that they would be working with the robot for six trials and that it would run two separate algorithms which would alternate between trials. All participants were told to abide by the following rules:

1) Treat the object as if it was a full glass of water (*to minimize rapid movements*).
2) Keep the object occluded in their grasp until letting go (*to keep final object position unknown until the end*).
3) Do not start the trajectory with the held object occluded underneath the table or with the arm resting on the table.
4) Do not place the next object down until the robot has returned to its ready pose.
5) Try to place the object down with a similar speed for all six trials (*in order to get the most unbiased results when comparing speed metrics*).

C. Metrics

Two sets of metrics were recorded to evaluate the effectiveness of our preemptive control pipeline. The first were quantitative metrics measured by the program which include response time, start-to-grab time, and prediction error. Response time was measured by the time it took for the robot to start a motion plan starting from its ready pose. Start-to-grab time was measured by the time it took for the robot to actually grab the object starting from its ready pose. Prediction error was measured as both the number of grids the model was off by as well as the euclidean distance in meters.

The second set of metrics were responses to the following questionnaire provided by the participants at the end of the experiment:

1) In terms of utilizing as much time as available, how efficient was each approach?
2) How fast was the response time for each approach?
3) How intelligent / intuitive did each approach seem?
4) How safe did you feel working with each approach?

Recorded responses consisted of a Likert score ranging from 1 to 5 for each approach for each question. Note that questions one (efficiency) and two (response time) were asked regardless of having access to quantitative measurements describing these two metrics. This was done in order to investigate how a human worker's perception of a robot coworker compared with measured results.
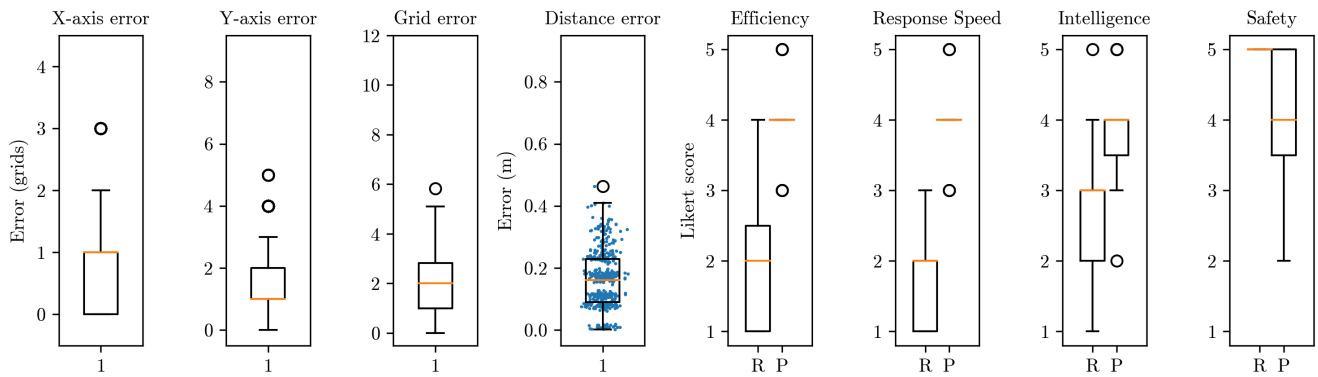
Fig. 6. The first four boxplots showcase the prediction errors in terms of number of x-grids, number of y-grids, number of combined (euclidean) grids, and euclidean error in meters. The error in meters boxplot has all samples plotted with noise to show the distribution. The last four boxplots showcase the Likert score results for the questionnaire from Sec. IV-C. Like Fig. 5, these subplots each have two boxplots, R for reactive and P for preemptive.

V. EVALUATION

A. Quantitative Results

The response and start-to-grab times for all grids and participants were boxplotted as shown in Fig. 5. The average response time improvement was 1.78 seconds with all eleven grid positions having a p-value < 0.0001 when comparing between their reactive and preemptive response times.

Still, the benefit of the reduced response times is largely dependent on the accuracy of our predictions. Here, the start-to-grab times can give a clearer view as to how much of the response time difference is actually used to move to an advantageous position. The average start-to-grab time improvement was 1.478 seconds which results in approximately 83% of the response time being utilized beneficially. Like response time, all eleven grid positions had a p-value < 0.0001 when comparing between reactive and preemptive start-to-grab times. Therefore, we conclude that preemptive planning largely benefits the human-to-robot indirect placement transfer task.

Finally, the prediction errors can be seen in the left four boxplots of Fig. 6. Note that the prediction errors plotted are the predictions used in the most recent predictive action before a definitive action was issued. Furthermore, as mentioned in Sec. III-F, a tolerance of about 2.236 euclidean grids was used to minimize stop-and-go stuttering. Our average prediction error was 2.025 euclidean grids which is below this tolerance.

B. Questionnaire Results

We define the questionnaire metrics as efficiency, response speed, intelligence, and safety for each of the questions from Sec. IV-C. The questionnaire results for each of these metrics can be seen plotted in the right four boxplots of Fig. 6. All questionnaire results had a p-value < 0.0001 when comparing between the reactive and preemptive responses. As shown, most participants found the preemptive approach to be more efficient and intelligent with faster response speeds when compared to the reactive approach. This can most likely be attributed to the noticeable reduction in task completion time as well as the concurrent motion of the robot with the participant.

Unsurprisingly, the only metric that the preemptive approach scored worse on was safety where the reactive approach received unanimous perfect scores. Interestingly, a few participants (even after giving positive evaluations about task efficiency and stating that they did not feel like they were ever in danger) said they preferred the reactive robot as it was more comfortable to work with. This highlights the foreignness and unfamiliarity of working closely with robots that some may possess. From these results, a clear trade-off between task efficiency and human comfort seems to exist. Therefore, choosing between the two will likely heavily depend on the task at hand as well as the personal preference of the human worker.

VI. CONCLUSIONS

In this paper, we introduced a novel preemptive control approach for human-to-robot indirect object placement transfers. This approach was shown to significantly reduce task completion time for indirect object placement transfers through a comprehensive case study between human participants and a robot manipulator. Furthermore, we concluded that preemptive robot behavior increases perceived efficiency and intelligence in exchange for general human comfort.

Future work involves improving the performance of our model and planner, conducting more ablation studies, as well as expanding the problem space. Although our model was able to correctly predict the general placement location a majority of the time, predictions still had a large variance as shown in Fig. 6. This can most likely be remedied by gathering more training data as well as possibly exploring non-recurrent-based architectures such as transformers. Additional ablation studies would also provide better insight of the intent model. In particular, although we used previous work [23] to help design our feature vector, a comparison between using just gestures versus both gestures & gaze would prove helpful. Finally, we would like to expand our preemptive control work to harder indirect placement transfer problems such as those involving a human walking from afar.

REFERENCES

[1] H. Nemlekar, D. Dutia, and Z. Li, "Object transfer point estimation for fluent human-robot handovers," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 2627–2633, 2019.

[2] W. Yang, C. Paxton, M. Cakmak, and D. Fox, "Human grasp classification for reactive human-to-robot handovers," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11123–11130, 2020.

[3] A. Doshi and M. M. Trivedi, "On the roles of eye gaze and head dynamics in predicting driver's intent to change lanes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 453–462, 2009.

[4] Y. Razin and K. Feigh, "Learning to predict intent from gaze during robotic hand-eye coordination," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[5] E. C. Townsend, E. Mielke, D. Wingate, and M. D. Killpack, "Estimating human intent for physical human-robot co-manipulation," *ArXiv*, vol. abs/1705.10851, 2017.

[6] U. Tariq, R. Muthusamy, and V. Kyrki, "Grasp planning for load sharing in collaborative manipulation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6847–6854, 2018.

[7] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1343–1350, 2017.

[8] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[9] M. Chen, S. Nikolaidis, H. Soh, D. Hsu, and S. Srinivasa, "Planning with trust for human-robot collaboration," in *Proceedings of ACM/IEEE International Conference on Human-Robot Interaction (HRI '18)*, pp. 307 – 315, February 2018.

[10] P. Kratzer, N. B. Midlagajni, M. Toussaint, and J. Mainprice, "Anticipating human intention for full-body motion prediction in object grasping and placing tasks," in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 1157–1163, 2020.

[11] S. Bansal, R. Newbury, W. Chan, A. Cosgun, A. Allen, D. Kulić, T. Drummond, and C. Isbell, "Supportive actions for manipulation in human-robot coworker teams," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11261–11267, 2020.

[12] H. S. Park, E. Jain, and Y. Sheikh, "Predicting primary gaze behavior using social saliency fields," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3503–3510, 2013.

[13] C. L. Nehaniv, K. Dautenhahn, J. Kubacki, M. Haegele, C. Parlitz, and R. Alami, "A methodological approach relating the classification of gesture to identification of human intent in the context of human-robot interaction," in *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.*, pp. 371–377, IEEE, 2005.

[14] J. Joo, W. Li, F. F. Steen, and S.-C. Zhu, "Visual persuasion: Inferring communicative intents of images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 216–223, 2014.

[15] L. Fan, W. Wang, S. Huang, X. Tang, and S.-C. Zhu, "Understanding human gaze communication by spatio-temporal graph reasoning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5724–5733, 2019.

[16] H. Admoni and B. Scassellati, "Social eye gaze in human-robot interaction: a review," *Journal of Human-Robot Interaction*, vol. 6, no. 1, pp. 25–63, 2017.

[17] A. Saran, S. Majumdar, E. S. Short, A. Thomaz, and S. Niekum, "Human gaze following for human-robot interaction," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8615–8621, IEEE, 2018.

[18] P. Kratzer, S. Bihlmaier, N. B. Midlagajni, R. Prakash, M. Toussaint, and J. Mainprice, "Mogaze: A dataset of full-body motions that includes workspace geometry and eye-gaze," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 367–373, 2020.

[19] M. Cakmak, S. S. Srinivasa, M. K. Lee, J. Forlizzi, and S. Kiesler, "Human preferences for robot-human hand-over configurations," in

[20] A. C. Huamán Quispe, E. Martinson, and K. Oguchi, "Learning user preferences for robot-human handovers," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 834–839, 2017.

[21] A. Moon, D. M. Troniak, B. Gleeson, M. K. Pan, M. Zheng, B. A. Blumer, K. MacLean, and E. A. Croft, "Meet me where i'm gazing: How shared attention gaze affects human-robot handover timing," in *Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '14, (New York, NY, USA), p. 334–341, Association for Computing Machinery, 2014.

[22] H. Admoni, A. Dragan, S. S. Srinivasa, and B. Scassellati, "Deliberate delays during robot-to-human handovers improve compliance with gaze communication," in *2014 9th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 49–56, 2014.

[23] P. Schydlo, M. Raković, L. Jamone, and J. Santos-Victor, "Anticipation in human-robot cooperation: A recurrent neural network approach for multiple action sequences prediction," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–6, 2018.

[24] A. Kshirsagar, M. Lim, S. Christian, and G. Hoffman, "Robot gaze behaviors in human-to-robot handovers," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6552–6558, 2020.

[25] M. K. Pan, E. A. Croft, and G. Niemeyer, "Evaluating social perception of human-to-robot handovers using the robot social attributes scale (rosas)," in *2018 13th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 443–451, 2018.

[26] I. D. Walker, L. Mears, R. S. M. Mizanoor, R. Pak, S. Remy, and Y. Wang, "Robot-human handovers based on trust," in *2015 Second International Conference on Mathematics and Computers in Sciences and in Industry (MCSI)*, pp. 119–124, 2015.

[27] K. Strabala, M. K. Lee, A. Dragan, J. Forlizzi, S. S. Srinivasa, M. Cakmak, and V. Micelli, "Toward seamless human-robot handovers," *J. Hum.-Robot Interact.*, vol. 2, p. 112–132, Feb. 2013.

[28] J. Mainprice and D. Berenson, "Human-robot collaborative manipulation planning using early prediction of human motion," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 299–306, 2013.

[29] R. Luo, R. Hayne, and D. Berenson, "Unsupervised early prediction of human reaching for human–robot collaboration in shared workspaces," *Autonomous Robots*, vol. 42, pp. 631–648, 2018.

[30] J. Mainprice, R. Hayne, and D. Berenson, "Predicting human reaching motion in collaborative tasks using inverse optimal control and iterative re-planning," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 885–892, 2015.

[31] L. Sun, W. Zhan, and M. Tomizuka, "Probabilistic prediction of interactive driving behavior via hierarchical inverse reinforcement learning," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2111–2117, 2018.

[32] Q. Wu, C.-J. Wu, Y. Zhu, and J. Joo, "Communicative learning with natural gestures for embodied navigation agents with human-in-the-scene," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[33] C.-M. Huang and B. Mutlu, "Anticipatory robot control for efficient human-robot collaboration," in *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 83–90, 2016.

[34] A. Jain, D. Chen, D. Bansal, S. Scheele, M. Kishore, H. Sapra, D. Kent, H. Ravichandar, and S. Chernova, "Anticipatory human-robot collaboration via multi-objective trajectory optimization," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11052–11057, 2020.

[35] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[36] R. P. Joshi, M. K. Broek, X. Z. Tan, A. Choi, and R. Luo, "Ros openpose." https://github.com/ravijo/ros_openpose, 2019.

[37] S. Lemaignan, F. Garcia, A. Jacq, and P. Dillenbourg, "From real-time attention assessment to "with-me-ness" in human-robot interaction," in *Proceedings of the 2016 ACM/IEEE Human-Robot Interaction Conference*, 2016.

[38] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "STOMP: Stochastic Trajectory Optimization for Motion Planning," in *International Conference on Robotics and Automation*, 2011.

2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1986–1993, 2011.