Proactive Robot Assistants for Freeform Collaborative Tasks through Multimodal Recognition of Generic Subtasks

Connor Brooks[1], Madhur Atreya[2], and Daniel Szafir[3]

Abstract— **Successful human-robot collaboration depends on a shared understanding of task state and current goals. In nonlinear or freeform tasks without an explicit task model, robot partners are unable to provide assistance without the ability to translate perception into meaningful task knowledge. In this paper, we explore the utility of multimodal recurrent neural networks (RNNs) with long short-term memory (LSTM) units for real-time subtask recognition in order to provide context-aware assistance during generic assembly tasks. We train RNNs to recognize specific subtasks in individual modalities, then combine the high-level representations of these networks through a nonlinear connection layer to create a multimodal subtask recognition system. We report results from implementing the system on a robot that uses the subtask recognition system to provide predictive assistance to a human partner during a laboratory experiment involving a human-robot team completing an assembly task. Generalizability of the system is evaluated through training and testing on separate tasks with some similar subtasks. Our results demonstrate the value of such a system in providing assistance to human partners during a freeform assembly scenario and increasing humans' perception of the robot's agency and usefulness.**

I. INTRODUCTION

Continued advances in robot design and affordability are enabling robots to expand beyond well-established factory-style robots to include robots designed for personal use and assistance in the home. This new avenue of robotics provides many challenges beyond those of the formulaic environment of factories, not the least of which is how to collaborate with human partners in unscripted tasks.

Collaboration in human teams relies on *common ground*: a shared base of knowledge among team members [15]. Humans use many mechanisms, both verbal and nonverbal, to establish common ground and coordinate actions while working together [5, 22]. In scenarios involving a task that is not predefined or is freeform in nature, one mechanism by which a robot might obtain common ground with its human partner is through recognition of the subtask that is in progress. For example, consider a robotic arm that provides assistance by handing tools to a human partner at the user's home workbench. As the workbench may be utilized for a variety of tasks, including furniture assembly, crafting, prototyping, or simply creative building, it may not be feasible for the robot to always have a formal representation of the high-level task being completed. However, if the robot can recognize when the human is performing subtasks that correlate to common assistance requests (e.g., the human is threading a nut onto a screw and the human often requests a screwdriver after threading a nut onto a screw), requests might be anticipated. Thus, subtask recognition could improve shared human-robot understandings of subtask phase, creating a more efficient partnership.

Subtask recognition has the potential to enable anticipatory robot actions, which prior research has shown can increase the fluidity of human-robot collaborations and improve human perceptions of the robot partner [4, 16, 17]. Additionally, since tasks of the same general type will share many of the same subtasks, subtask recognition provides a generalizable approach to gaining common ground. Recognizing and responding to generic subtasks has the potential to allow the implementation of anticipatory robot assistants in unstructured environments lacking well-defined tasks, where prior strategies that utilize *a priori* knowledge of the task structure are often not feasible or effective.

In this paper, we design a system for fusing multimodal data from commercial-off-the-shelf sensors to create a subtask recognition system. The system is based on recurrent neural networks (RNNs) with long short-term memory (LSTM) cells, allowing it to be trained on any labeled data without hand-coded feature selection. We train the system on data collected from participants completing an assembly task, then evaluate the system in a laboratory experiment in which participants work with a robot on a separate task involving similar subtasks in order to investigate generalizability.

II. BACKGROUND

Our approach draws on prior work related to coordination in human-robot teaming, human activity recognition (HAR), and multimodal machine learning. In this section, we review the developments in these fields that inspire our work.

A. Coordination in Human-Robot Teaming

Previous work on coordination in human-robot teaming has taken theories from cognitive science and related disciplines and applied them to human-robot interaction. Hoffman and Breazeal [15] describe the importance of common ground for the completion of joint actions. Following this underlying theory, Mutlu et al. [22] compares methods for using robot behavior to invoke different coordination mechanisms for humans and robots involved in joint action (actions involving collaboration between partners). These coordination mechanisms help the human and robot gain common ground and coordinate their actions to achieve a shared goal.

[1]Department of Computer Science, University of Colorado Boulder. connor.brooks@colorado.edu
[2]Department of Mechanical Engineering, University of Colorado Boulder. madhur.atreya@colorado.edu
[3]Department of Computer Science and ATLAS Institute, University of Colorado Boulder. daniel.szafir@colorado.edu

Various methods of coordination in human-robot teams have been investigated. These include environment state factors, such as using contextual knowledge of the environment to improve understanding of referring expressions [29]. Other methods of coordination include those focusing on the human partner, such as eye tracking to predict human requests from a set of discrete choices [17] and modeling human behavior types to choose a proper interaction scheme [18, 23].

Additionally, multiple studies have investigated the use of a task status estimator or future task state predictor for anticipatory assistance behavior. Hawkins et al. [12] and Baraglia et al. [4] both use a probabilistic graphical model to estimate task status based on tracked hand and object positions in order to plan for possible assistance actions. Similar goals have been achieved using finite-state machine task models operating on hand-coded features [10].

While these systems demonstrate various methods for obtaining common ground in joint actions, they do not demonstrate generalizability beyond the task for which they were designed. Additionally, each of these systems assumes knowledge of the current task structure or the distribution of subtask steps needed for completion. As mentioned previously, however, this information may not be available for the integration of robots in environments involving freeform tasks.

B. Human Activity Recognition

Activity recognition involves classification of human activities from various types of data. The problem of subtask recognition is fundamentally an activity recognition problem.

There are various approaches to activity recognition, such as estimating a hidden state that represents human intention [30], using probabilistic graphical models [2, 28], applying a Support Vector Machine on key poses [6] and fitting Gaussian Mixture Models over human trajectories [13].

With the success of deep learning in the last decade drawing attention to the field, many researchers have investigated the use of deep learning for activity recognition. Baccouche et al. [3] were the first to apply a combination of convolutional and recurrent neural network models to the KTH human activity dataset, achieving benchmark results. Ordóñez et al. [25] and Donahue et al. [8] also use convolution-recurrent networks for classification of activities using wearable sensors and video sequences, respectively. Du et al. [9] use layered RNNs on skeletal position data for classifying activities.

Due to the recent state-of-the-art results provided by RNN-based activity recognition systems and the natural fit of RNNs for classifying ongoing sequential data at each new timestep, we chose to explore the use of RNN-based classifiers for subtask recognition.

C. Multimodal Machine Learning

In order to recognize subtasks of various types, we use a multimodal classification system. Such systems have been shown to provide improved performance over single-modality systems through the specialization of various input modalities for recognizing some subset of possible classes [11]. We follow a common approach in usage of multimodal classifiers that involves combining classifications from separate modalities through a nonlinear fusion technique [7, 19].

III. TRAINING DATASET

To obtain training data for our system, we ran a study that involved recording participant dyads completing an assembly task. Each dyad was split into an *assembler* and an *assistant*, with the assistant role filled by an experimental confederate (a researcher playing the part of a participant, unknown to the assembler). This setup was chosen to imitate collaboration in a human-robot team, with the confederate taking the place of a robot assistant that helps hand tools and provide assembly assistance to a primary assembler.

The existing body of human-robot interaction literature includes several different assembly tasks ranging from the stacking of blocks [1] to the simple installation of fasteners [20]. For our study, we sought to develop a task that was ecologically valid to a generic assembly setting, such as an individual building a small piece of furniture from scratch. With this in mind, we designed and constructed a small wooden assembly that did not resemble any familiar product.

A. Data Recording

Recording was done using commercial off-the-shelf technology that imitates the sensor technology available on many mobile manipulator robots. The sensors used were a Microsoft Kinect V2 and a USB microphone. Frames were captured from the Kinect at a rate of 15 Hz. Audio was sampled at a rate of 44.1 kHz and divided into sequences 0.067 seconds in length to provide an equal number of image frames and audio sequences.

The assembler and assistant stood on opposite sides of the table, while the camera was set up next to the assistant and facing the assembler in order to mimic the view of a robot assistant positioned across the table from its human partner.

The training data collection involved a total of 16 participants (8 male and 8 female). Task completion time ranged from 315 seconds to 907 seconds. Total in-task time across all dyads was approximately 9,545 seconds. Frames were recorded at a rate of 15Hz, giving 143,175 frames of data.

B. Feature Extraction

From this data, we derived three data input modalities for our system: body pose, body movement, and audio features. For audio features, we used a spectrogram created from the log magnitude of 40 bins divided according to the mel scale of frequency. The mel scale mimics human perception of sound by mapping frequency to pitch as perceived by humans, which includes more dense measures at lower pitches and sparser divisions at frequencies above 1 kHz [21].

The Kinect API was used to extract body pose information of the assembler from each frame. We used the location of 9 joints comprising the torso and upper arms (lower arm data might have been useful for classifying subtasks, but was not used as the assembler's lower arm was often occluded and the lower arm tracking data was extremely noisy when working

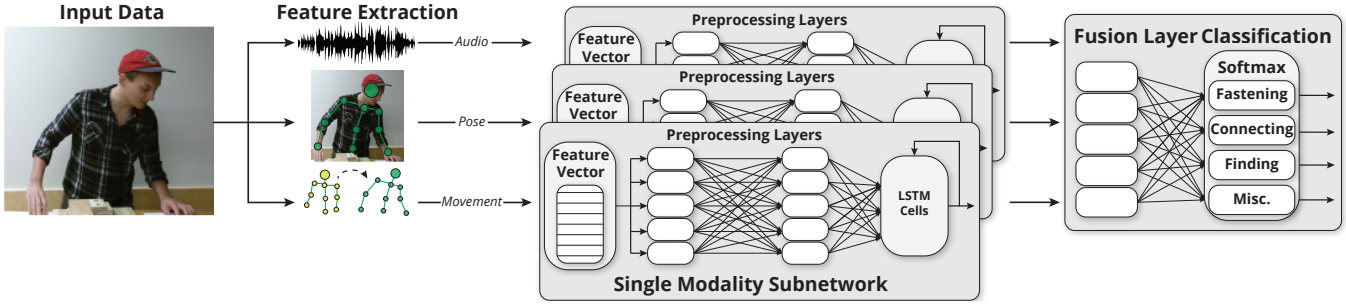

Fig. 1. The full classification system architecture.

with tools and moving in and around the assembly). From these locations, we extracted the relative orientation of each joint compared to its parent joint, as well as the movement of each joint position between consecutive frames.

Each of these three feature types have previously been used in modeling activities or events [2, 9, 11, 26, 30]. Other potential features that we chose not to use include image frames and absolute body position. These features were not used in order to emphasize features that are more generalizable across task, individual participant, and environment setup. For example, body position and background image vary with camera placement and subject workstation positioning, but pose and movement should not as these features are relative to the position and angle of the recorded person.

C. Video Annotation

After data collection was completed, the videos were annotated for various subtasks. Two coders annotated the videos with an overlap of 14.5% of the dataset coded by both. Inter-rater reliability analysis demonstrated substantial agreement between the coders (Cohen's κ = .716). Labels for training our model were dynamically generated from these video annotations. The codes annotated on the videos can be broken down as follows.

1) Fastening: One code related to the subtask of fastening: Actively fastening a screw and tightening washers on a screw. Both of these activities involve the same basic motions and goals, and were consequently grouped together for subtask recognition purposes.

2) Connecting Parts: The next group of codes involved the connection of two pieces during assembly. This was broken into two separate codes: aligning two pieces (typically involving aligning screw holes) and inserting a screw or rod to connect the pieces.

3) Finding Parts: Another group of codes involved searching for (often small) parts. The two codes involved in this group were sorting through a group of parts and identification of a part (typically done by holding a part up to one's face).

4) Miscellaneous: Finally, three other codes were annotated that did not fit into the three former subtask groups. These codes were requesting help, adding information to a help request, and completion of a subtask.

IV. MULTIMODAL SUBTASK RECOGNITION

A. Long Short-Term Memory Recurrent Neural Networks

RNNs provide a mechanism for classifying sequences through retaining a memory state that is updated with every new input. While being trained, RNNs are "unrolled" through time to backpropagate over the sequence. Unrolling is done through creating separate network layers for each timestep in order to propagate error from the most recent output back to the beginning of the sequence. Thus, RNNs are *deep in time*.

LSTM units improve RNNs by creating a structure for maintaining memory over time while learning what information to keep and what information can be forgotten. This is done through gates that control input (**i**), output (**o**), and forgetting (**f**) within the LSTM unit. LSTM units have been shown to reduce the effects of the vanishing and exploding gradients that made RNNs unable to properly backpropagate error over long sequences. This improvement is due to the ability of LSTMs to maintain a more constant error throughout the sequence [14]. Consequently, LSTM-RNNs are able to learn long-term dependencies within data sequences that RNNs were previously unable to successfully learn. We use the LSTM formulation described by Hochreiter and Schmidhuber [14] and used in other activity recognition systems [9, 19, 25]:

$$\mathbf{i}_t = \sigma(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{W}_{ci}\mathbf{c}_{t-1} + \mathbf{b}_i) \quad (1)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{W}_{cf}\mathbf{c}_{t-1} + \mathbf{b}_f) \quad (2)$$

$$\mathbf{c}_t = \mathbf{f}_t\mathbf{c}_{t-1} + \mathbf{i}_t * tanh(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c) \quad (3)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{W}_{co}\mathbf{c}_t + \mathbf{b}_o) \quad (4)$$

$$\mathbf{h}_t = \mathbf{o}_t * tanh(\mathbf{c}_t) \quad (5)$$

where $\mathbf{i}_t$ is the input gate, $\mathbf{f}_t$ is the forget gate, $\mathbf{c}_t$ is a memory cell that retains information from previous inputs, $\mathbf{o}_t$ is the output gate, and $\mathbf{h}_t$ is the hidden unit that is passed as input to the next timestep. LSTM units retain information through the use of memory cell $\mathbf{c}$ and hidden state $\mathbf{h}$. Multiple LSTM units can be stacked using the $\mathbf{h}$ values at each layer as inputs to the next layer.

B. Network Structure

We used separate RNNs for each input modality, allowing for individual modality tuning of hyperparameters such as network architecture. For each RNN, data from the respective modality was first fed through two feedforward layers to

TABLE I

RESULTS FROM LEAVE-ONE-OUT VALIDATION ON THE TRAINING DATA USING $F1$ SCORES WITH THE BEST PERFORMING NETWORK ON EACH TASK IN BOLD.

	Task				
Network	Null Frames	Fastening	Connecting	Finding	Mean
Audio	0.483	0.573	0.261	0.299	0.404
Pose	0.578	0.573	0.245	0.260	0.414
Movement	0.607	0.621	**0.331**	0.308	0.467
Multimodal	**0.632**	**0.626**	0.294	**0.349**	**0.475**

preprocess the inputs to the LSTM units. Then, the output from the feedforward layers was fed into an LSTM unit. This LSTM unit contained multiple layers and varying layer sizes. We tuned these hyperparameters for each modality while keeping the size of the last layer consistent between networks: the audio network used a single layer of size 8, while the body and pose networks both used a three-layer LSTM unit with layer sizes 64, 16, and 8. To test individual modality classification, the output of the LSTM unit was passed through a single feed-forward layer into a softmax layer to obtain a classification.

The multimodal classifier combined the outputs from each LSTM unit in a single connection layer, then passed the outputs of the connection layer into a final softmax layer. This strategy allows for a nonlinear fusion of data modalities, with the fusion layers being trained on the high-level representations from each modality. This technique of using a nonlinear combination of individual modalities has been shown to outperform concatenation of different feature types and linear combination of classifications on similar problems involving fusion of multiple sensory streams [19].

C. Validation Results

Table I presents the results from a leave-one-out validation test on the training data over the individual participants. These results use 10-second sequences of data, with the final classification in the sequence compared to the label for the final frame in the sequence. Each possible 10-second sequence within the training data is tested.

Results are presented using the $F1$ score of the classifier on the validation set. This score is given by $F1 = 2 * \frac{P*R}{P+R}$, where $P =$ the *precision* of the classifier on the respective task and $R =$ the *recall*. As some of the tasks occur infrequently in the dataset, these numbers provide a better metric of the classifier's recognition of the subtask than raw accuracy.

Results are given for each of the aforementioned subtasks, as well as null frames (any frame that is not labeled as either fastening, connecting parts, or finding parts). The mean $F1$ score is reported for each classifier as well.

V. EXPERIMENTAL EVALUATION

To evaluate our system, we ran an experiment with a total of 14 participants (10 males, 4 females) recruited from around the University of Colorado Boulder campus and surrounding community. The average participant age was 22.9 ($SD = 8.02$), ranging from 18–49. There was no overlap between participants used for collecting training data and the participants in this experiment. The experiment took approximately 30 minutes, and participants were paid $5.00 for their time.

The experimental procedure involved completing an assembly task consisting of several subtasks with the help of a 6-DOF Kinova Jaco robotic arm that delivered cups of parts. The assembly on which the participant worked was of similar size to that which was used in the training data, but was a different shape and had different types of components. Likewise, the camera angle was altered and the lighting also adjusted. These alterations were to confirm that the features fed into our system were not highly reliant on precise conditions. A comparison of the two experiment setups can be seen in Fig. 2.

The experiment consisted of 20 subtasks divided evenly between 4 categories. The subtask categories were set up as follows:

1) Screwing Task: Participants were instructed to fasten the pair of screws at one of the five screwing slots. One screw was already in place; participants had to first fasten this screw, then they had to get the cup of screws from the robot in order to retrieve the second screw for this slot. This subtask is analogous to the "fastening" subtask described in Section III-C.1.

2) Pegging Task: Participants were instructed to line up a block containing three holes with a set of holes on the assembly structure and place two pegs in the outer two holes to hold the block in place. Then, participants had to get the cup of black pegs from the robot and place a black peg in the center hole. This subtask is analogous to the "connecting parts" subtask described in Section III-C.2.

3) Component Task: Participants were instructed to find the electrical component identified by a specified four-character label from a cup of electrical components. Once the participant found the component, the participant had to get the cup containing tape from the robot and tape the component in the specified slot. This subtask is analogous to the "finding parts" subtask described in Section III-C.3.

4) Wiring Task: Participants were instructed to place a jumper wire on the assembly's breadboard, connecting the two specified breadboard coordinates. No robot assistance was needed for this task. This subtask is not analogous to any of the trained subtasks and acts as the null subtask. This subtask was included to represent that robots may not always be able to provide assistance and will encounter subtasks which they have not been trained to recognize.

A. Experimental Design and Procedure

Our experiment took the form of a 2×1 between-participants design with participants randomly assigned to conditions. The main independent variable represented the type of underlying robotic system. Participants interacted with either a *command-driven assistant* or *proactive assistant*.

In the command-driven assistant condition, participants completed each subtask and then requested robot assistance

(if needed) through buttons on a simple iPad interface. The robot would only take action in response to these direct participant requests.

In the proactive assistant condition, the multimodal classification system described in Section IV made classifications on the sequence of frames within a subtask in real time. Participants in the proactive assistant condition still had the option of requesting parts directly using the iPad in case the robot either had not made a classification or had made an incorrect classification.

In both the command-driven assistant and proactive assistant conditions, the participant was prevented from requesting robot assistance until 20 seconds had passed since the start of the subtask. This time constraint was adopted to prevent the participants from immediately requesting the next part needed before completing the first step of the subtask.

The classification system used thresholds that were chosen based on preliminary data from two pilot runs of the experiment in order to determine an activation level for each subtask (note that the network was not altered in any way from this pilot data). These thresholds were chosen empirically through observing the activation levels of each classification throughout the task. During the experiment, if the softmax output of a particular classification subtask reached its threshold value, the system fired a command to the robotic arm to preemptively offer the cup of parts needed for the correlated experiment subtask. No threshold was set for the null task; the correct action for the null task was for the system to not fire on any of the other subtasks. The system made a maximum of one guess for each subtask: once it fired on any particular subtask, the classification stopped and waited until it received notification that a new subtask had begun. The subtask classification system did not retain any information between subtasks; the internal states of all LSTM units were reset at the beginning of each new subtask and no information about previously completed subtasks was recorded or used by the system.

B. Measures

We used both objective and subjective measures to evaluate our approach. Objective measures included *actuated classification accuracy*: the accuracy of active classifications the robot made to help the users (subtasks in which the robot made no classification not counted), and *overall accuracy*: total classification accuracy for participants comprising both correctly classified robot actions and correctly classified times when the robot chose not to act.

We also collected data from participants on their subjective perceptions of the robot. We followed the standard scale construction methodology [27] to create two scales using 7-point Likert-style questionnaire items that measured perceived *robot agency and initiative* (5 items, Cronbach's α = 0.931) and *robot usefulness* (5 items, Cronbach's α = 0.916).

Fig. 2. The image on the left shows a training frame classified as the "connecting parts" subtask, while the image on the right shows a test experiment participant completing the analogous pegging task. See the video accompaniment for more examples.

VI. RESULTS

A. System Accuracy

Through the 7 proactive assistant-condition participants, the average actuated classification accuracy (as defined in V-B) was 0.577 (SD = 0.203), with a recall rate of 0.362 on the non-null subtasks. The average overall accuracy for the system on these 7 participants, including null subtasks (counted as correct if no classification was made, incorrect otherwise), was 0.424 (SD = 0.085). The accuracy level is higher for the actuated classifications because the system defaults to null if it is not confident enough to surpass the thresholds set for each task, thus there are fewer false positives at the cost of lower recall.

We also conducted a *post-hoc* analysis of the system's classifications on all participants' data from both conditions. These classifications were not actuated for the participants in the command-driven assistant condition, but were still recorded for analysis purposes. For all participants, the average accuracy on all actuated classifications made was 0.483 (SD = 0.227). The average overall accuracy (including null subtasks) was 0.374 (SD = 0.136). As the standard deviations of these results show, the system's performance varied heavily based on individual. The classification accuracies are shown in Fig. 3 (b).

B. Subjective Results

We analyzed our subjective results using a one-way analysis of variance (ANOVA) with the experimental condition as a fixed effect, as parametric tests (e.g., ANOVA) are robust for constructed scales [24]. We report the F-statistic and degrees of freedom for these tests. Our results are visualized in Fig. 3. We found a significant effect of the experimental condition on perception of robot agency and initiative, $F(1, 12) = 57.79$, $p < 0.0001$, where participants rated the proactive assistant ($M = 3.69$) as having more agency and initiative than the command-only assistant ($M = 1.43$), visualized in Fig. 3 (a).

However, our analysis did not reveal a significant effect of the experimental condition on perceptions of robot usefulness, $F(1,12) = 0.033$, $p = 0.859$, as seen in Fig. 3 (c). To better understand the link between perceived usefulness, proactive assistance, and classification accuracy, we used regression analysis to test effects of the ratio of correct to incorrect predictive assistance actuations and perception of robot usefulness among participants in the proactive assistant condition. The results of the regression indicated actuated

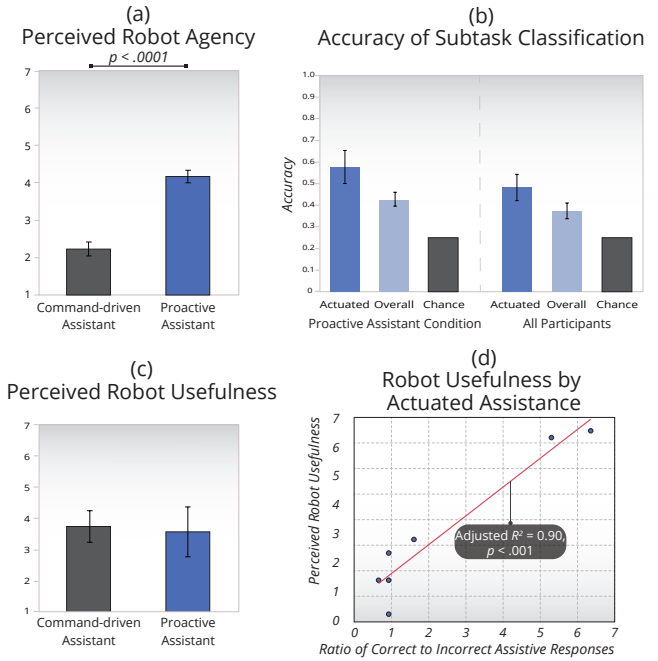

Fig. 3. Our subjective results demonstrate the impact of condition on perception of robot agency and initiative, the correlation between the ratio of correct to incorrect predictive assistance actuations and usefulness. Additionally, our objective results demonstrate the accuracy of the system compared to the baseline performance due to chance.

assistance accuracy was a strong predictor of perceived robot usefulness, explaining 91.7% of the observed variance (Adjusted $R^2 = 0.900$, $F(1, 5) = 55.1$), as shown in Fig. 3 (d). This result highlights the strong potential for negative effects of incorrect classifications on the participant's perception of the robot.

VII. Discussion and Conclusions

In this paper, we have presented a method for providing predictive assistance in freeform tasks through subtask recognition that generalizes beyond the task on which it was trained. While the accuracies provided by our system are variable based on participant, the system performs well above the baseline on average. An experimental evaluation showed that the system increased participants' perception of the robot's agency and that user perception of robot usefulness was closely correlated with classification accuracy.

While our method for providing assistance in freeform tasks demonstrates a path toward developing more general-purpose helper robots, further work could improve integration of such a system on a general-purpose robot. Although our system provides contextual awareness, integrating this knowledge with action would require an additional step of correlating subtasks with helping action, perhaps by tracking what type of help users most often request after each observed subtask. The finding from our experiment of the high variance in system accuracy among participants suggests several exciting avenues for further work. For instance, our system might be improved by clustering users based on initial interaction to anticipate or customize particular user models.

Overall, this system demonstrates a promising method for developing fully automated systems that are capable of providing generalized predictive assistance in real time based on subtask recognition. Additionally, our results on the perception of robot usefulness provide guidance for the designers of interactive robotic systems through demonstrating the strong potential for harmful effects of incorrect or undesired anticipatory action on the human partner's perception of the robot teammate, suggesting that future systems prioritize a minimization of false positives, rather than necessarily optimizing for true positives. We believe our method of predictive assistance has the potential to advance the capabilities of robot partners in environments with relaxed constraints that pose challenges to more traditional techniques.

Acknowledgment

This work was supported by an NSF CRII Award #1566612. We thank Adrienne Stenz, Parker Steinberg, Andrew Gorovoy, and Radhen Patel for their help with this research.

References

[1] Henny Admoni, Thomas Weng, Bradley Hayes, and Brian Scassellati. Robot Nonverbal Behavior Improves Task Performance in Difficult Collaborations. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI '16)*, pages 51–58. IEEE, 2016.

[2] Mohammad M Arzani, Mahmood Fathy, Hamid Aghajan, Ahmad A Azirani, Kaamran Raahemifar, and Ehsan Adeli. Structured Prediction with Short/Long-Range Dependencies for Human Activity Recognition from Depth Skeleton Data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '17)*, pages 560–567. IEEE, 2017.

[3] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. Sequential Deep Learning for Human Action Recognition. In *International Workshop on Human Behavior Understanding*, pages 29–39. Springer, 2011.

[4] Jimmy Baraglia, Maya Cakmak, Yukie Nagai, Rajesh Rao, and Minoru Asada. Initiative in Robot Assistance During Collaborative Task Execution. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI '16)*, pages 67–74. IEEE Press, 2016.

[5] Cynthia Breazeal, Cory D Kidd, Andrea Lockerd Thomaz, Guy Hoffman, and Matt Berlin. Effects of Nonverbal Communication on Efficiency and Robustness in Human-Robot Teamwork. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '05)*, pages 708–713. IEEE, 2005.

[6] Enea Cippitelli, Samuele Gasparrini, Ennio Gambi, and Susanna Spinsante. A Human Activity Recognition System using Skeleton Data from RGBD Sensors. *Computational Intelligence and Neuroscience*, 2016:21, 2016.

[7] Francisco Cruz, German I Parisi, Johannes Twiefel, and Stefan Wermter. Multi-Modal Integration of Dynamic Audiovisual Patterns for an Interactive Reinforcement

Learning Scenario. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '16)*, pages 759–766. IEEE, 2016.

[8] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pages 2625–2634, 2015.

[9] Yong Du, Wei Wang, and Liang Wang. Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pages 1110–1118, 2015.

[10] Hiraki Goto, Jun Miura, and Junichi Sugiyama. Human-Robot Collaborative Assembly by On-line Human Action Recognition Based on an FSM Task Model. In *Workshop on Collaborative Manipulation, ACM/IEEE International Conference on Human-Robot Interaction (HRI '13)*, 2013.

[11] Fei Han, Xue Yang, Christopher Reardon, Yu Zhang, and Hao Zhang. Simultaneous Feature and Body-Part Learning for Real-Time Robot Awareness of Human Behaviors. In *IEEE International Conference on Robotics and Automation (ICRA '17)*, pages 2621–2628. IEEE, 2017.

[12] Kelsey P Hawkins, Shray Bansal, Nam N Vo, and Aaron F Bobick. Anticipating Human Actions for Collaboration in the Presence of Task and Sensor Uncertainty. In *IEEE International Conference on Robotics and Automation (ICRA '14)*, pages 2215–2222. IEEE, 2014.

[13] Bradley Hayes and Julie A Shah. Interpretable Models for Fast Activity Recognition and Anomaly Explanation During Collaborative Robotics Tasks. In *IEEE International Conference on Robotics and Automation (ICRA '17)*, pages 6586–6593. IEEE, 2017.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

[15] Guy Hoffman and Cynthia Breazeal. Collaboration in Human-Robot Teams. In *AIAA 1st Intelligent Systems Technical Conference*, page 6434, 2004.

[16] Guy Hoffman and Cynthia Breazeal. Effects of Anticipatory Action on Human-Robot Teamwork Efficiency, Fluency, and Perception of Team. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI '07)*, pages 1–8. ACM, 2007.

[17] Chien-Ming Huang and Bilge Mutlu. Anticipatory Robot Control for Efficient Human-Robot Collaboration. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI '16)*, pages 83–90. IEEE Press, 2016.

[18] Chien-Ming Huang, Maya Cakmak, and Bilge Mutlu. Adaptive Coordination Strategies for Human-Robot Handovers. In *Robotics: Science and Systems (RSS '15)*, 2015.

[19] Ashesh Jain, Avi Singh, Hema S Koppula, Shane Soh, and Ashutosh Saxena. Recurrent Neural Networks for Driver Activity Anticipation via Sensory-Fusion Architecture. In *IEEE International Conference on Robotics and Automation (ICRA '16)*, pages 3118–3125. IEEE, 2016.

[20] Przemyslaw A. Lasota, Gregory F. Rossano, and Julie A. Shah. Toward Safe Close-Proximity Human-Robot Interaction with Standard Industrial Robots. In *IEEE International Conference on Automation Science and Engineering (CASE '14)*, pages 339–344. IEEE, 2014.

[21] Beth Logan. Mel Frequency Cepstral Coefficients for Music Modeling. In *ISMIR*, volume 270, pages 1–11, 2000.

[22] Bilge Mutlu, Allison Terrell, and Chien-Ming Huang. Coordination Mechanisms in Human-Robot Collaboration. In *Workshop on Collaborative Manipulation, ACM/IEEE International Conference on Human-Robot Interaction (HRI '13)*, pages 1–6. Citeseer, 2013.

[23] Stefanos Nikolaidis, Ramya Ramakrishnan, Keren Gu, and Julie Shah. Efficient Model Learning from Joint-Action Demonstrations for Human-Robot Collaborative Tasks. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI '15)*, pages 189–196. ACM, 2015.

[24] Geoff Norman. Likert Scales, Levels of Measurement and the "Laws" of Statistics. *Advances in Health Sciences Education*, 15(5):625–632, 2010.

[25] Francisco Javier Ordóñez and Daniel Roggen. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *Sensors*, 16(1):115, 2016.

[26] Giambattista Parascandolo, Heikki Huttunen, and Tuomas Virtanen. Recurrent Neural Networks for Polyphonic Sound Event Detection in Real Life Recordings. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '16)*, pages 6440–6444. IEEE, 2016.

[27] Paul E Spector. *Summated Rating Scale Construction: An Introduction*. Number 82. 1992.

[28] Zhikun Wang, Katharina Mülling, Marc Peter Deisenroth, Heni Ben Amor, David Vogt, Bernhard Schölkopf, and Jan Peters. Probabilistic Movement Modeling for Intention Inference in Human-Robot Interaction. *The International Journal of Robotics Research*, 32(7):841–858, 2013.

[29] David Whitney, Eric Rosen, James MacGlashan, Lawson LS Wong, and Stefanie Tellex. Reducing Errors in Object-Fetching Interactions through Social Feedback. In *IEEE International Conference on Robotics and Automation (ICRA '17)*, pages 1006–1013. IEEE, 2017.

[30] Lu Xia, Chia-Chih Chen, and JK Aggarwal. View Invariant Human Action Recognition using Histograms of 3D Joints. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '12)*, pages 20–27. IEEE, 2012.