Interactive Task Learning for Social Robots: A Pilot Study

Alexander Tyshka[1] and Wing-Yue Geoffrey Louie[1]

***Abstract*— For socially assistive robots to achieve widespread adoption, the ability to learn new tasks in the wild is critical. Learning from Demonstration (LfD) approaches are a popular method for learning in the wild, but current methods require significant amounts of data and can be difficult to interpret. Interactive Task Learning (ITL) is an emerging learning paradigm that aims to teach tasks in a structured manner, minimizing the need for data and increasing transparency. However, to date ITL has only been explored for physical robotics applications. Additionally, minimal research has explored how usable existing ITL systems are for non-expert users. In this work, we propose a novel approach to learn social tasks via ITL. This system utilizes recent advances in Natural Language Understanding (NLU) to learn from natural dialogue. We conducted a pilot study to compare the ITL system against an LfD approach to investigate differences in teaching performance as well as teachers' perceptions of trust and workload towards these systems. Additionally, we analyzed the teaching behavior of participants to identify successful and unsuccessful teaching strategies. Our findings suggest ITL could provide more transparency to users and improve performance by correcting speech recognition errors. However, participants generally preferred LfD and found it an easier teaching method. From the observed teaching behavior, we identify existing challenges in ITL for non-experts to teach social tasks. Using this, we propose areas of improvement toward future ITL learning paradigms that are intuitive, transparent, and performant.**

I. Introduction

Socially assistive robots (SARs) have tremendous potential to improve our society, yet in order to do so these robots require a means of learning how to interact with humans in different tasks and settings. Given the infeasibility of designing a fully general robot, it is imperative that non-expert users can teach and adapt SARs in the wild. A popular approach for this is learning from demonstration (LfD), where a human demonstrates a task and the robot forms a model that is used to execute the task independently. LfD has shown promising results in physical domains such as object manipulation as well as social domains such as therapy for Autism Spectrum Disorder [1] and group activities for older adults [2]. However, it can be difficult to teach tasks to SARs because while these robots may look human they do not have human-level cognition. Teachers may overestimate the reasoning or common-sense knowledge of the robot based on its humanoid appearance. This is referred to as the perceptual belief problem [3]. It can significantly impair LfD because a teacher cannot be effective without understanding what concepts the robot already knows and what it needs to learn. For SARs to achieve greater autonomy, they must be able to rapidly acquire new concepts and convey the extent of their knowledge to their teachers.

Interactive task learning (ITL) is a new learning paradigm that seeks to address this problem [4]. ITL expands LfD to include natural language and demonstration, aiming to mimic human learning and more closely integrate the teacher. While LfD treats the human as an actor or an expert, ITL treats the human as a teacher who explains the nature of the task and breaks it down into learnable components. By using natural language, the teacher can convey knowledge more efficiently as well as provide a grounding for learned concepts that the robot can then use to explain its knowledge [5].

Recent works have explored the use of ITL in manipulation tasks [4], [6], [7] but ITL has strong potential to address open challenges in social robotics. Data-efficiency is one such challenge for SARs because social interactions are not easily simulated and require real world data. In a rich social environment, robots must learn what features to focus on. This requires crafting feature sets ahead of time (reducing generalization capability) or learning from raw data using deep neural networks, which can struggle on limited training data. Using natural language to teach social robots could significantly reduce the demonstration data needed because concepts, rules, and constraints of the task can be directly described rather than inferred from a large dataset.

However, existing ITL approaches used in physical HRI are not readily applicable for teaching social HRI. In a manipulation task, learned concepts usually correlate a word with a physical object, attribute, or action. In ITL these concepts are often taught by focusing the robot's attention using gestures [8] or physically demonstrating actions. In most manipulation experiments (e.g. [6], [8]), the environment has a finite set of objects present, which effectively constrains the vocabulary used. In contrast, social tasks are often abstract and difficult to ground. States and actions are based on the dialogue and not the physical environment. Therefore concepts must be learned verbally and without the aid of physical teaching cues. The teaching vocabulary can be much more open-ended without a physical environment to constrain it. To provide a reasonable response, a social ITL system must have robust parsing that can understand a wide range of commands and vocabulary, which the handcrafted parsers typically used in ITL may struggle with. Finally, social tasks may be more difficult to teach to robots than physical tasks. While physical tasks or games are often intuitive to break down into rules, steps, and sub-tasks, social tasks can be much less structured and rely on human intuition and common sense. To use ITL for social tasks, it must

*This work was supported by the National Science Foundation grant #1948224 and #2238088

[1]Intelligent Robotics Laboratory, Oakland University, Michigan, USA, 48309 (e-mail: louie@oakland.edu, atyshka@oakland.edu)

address this challenge and induce computational thinking in the human during the learning process.

Additionally, there is a current research gap of understanding how end-users use and perceive existing ITL systems. How intuitive is teaching with ITL? What mental models do teachers form about the robot? Do they prefer this learning paradigm over others like LfD? Some research has investigated human teaching behavior in Wizard-of-Oz studies [9] as well as virtual studies identifying failure cases [10], but to the best of our knowledge no existing HRI studies investigate fully autonomous ITL systems with non-expert users. To realize the full potential of ITL for social robots, we must evaluate such systems with non-expert users to investigate how performant, intuitive, and transparent these systems are.

In this work, we present a preliminary approach to learn social tasks via ITL and evaluate user perceptions of this system with an HRI study. This approach uses recent advances in natural language processing to adapt to a range of unstructured language without the need for extensive handcrafted rules. A learning agent guides the human teacher through ITL while attempting to induce computational thinking. Our HRI study compares this system against a pure LfD baseline on the post-teaching robot performance as well as participant trust and perceived workload. Using feedback from participants and observations from both LfD and ITL teaching sessions, we identify areas for improving the intuitiveness and transparency of learning systems for SARs.

II. Related Works

Given the wide range in applications, LfD has been used in many works for teaching social robots [1], [2], [11]. However, demonstrating tasks can take significant time, especially in complex environments where much data is required to separate patterns from noise. Language-conditioned learning seeks to address this problem by generating novel robot behavior from a verbal command. It has shown great success in physical manipulation areas [12], [13], enabling robots to execute complex action sequences in real and simulated environments from language commands.

While LfD and language-conditioned learning have been successfully utilized in numerous robotics tasks, many approaches use an end-to-end neural network design that can inhibit interpretability and generalization. However, interpretability is especially important for social robots, where inappropriate behavior can be particularly detrimental. Global interpretability helps teachers understand what has been learned, and local interpretability can improve accuracy and user trust by rationalizing individual decisions [14]. While a number of works [15], [16] have explored interpretability in LfD, a conflict arises between generalization and explanation quality. High-level, natural language explanations, as used in [17], are most suitable for non-expert users. However, these approaches predominantly use end-to-end neural techniques and can require thousands of labeled explanations of a task. The end-to-end design also prevents knowledge transfer between models, as there is no explicit modeling of concepts, only a latent space. Additionally, because these explanations are labeled and trained offline, such methods cannot provide interpretability while teaching the robot. Alternative approaches [15] use inherently-interpretable models, but do not explain in natural language or high-level concepts, and therefore are more suitable for expert users.

Given the weaknesses of purely neural methods for in-situ learning, hybrid approaches combining machine learning and structured models provide a promising alternative. Walker et al. [18] propose a language-conditioned learning approach that utilizes an intermediate logical grammar to enable interpretability and generalization to unseen tasks. Mao et al. [19] present a neuro-symbolic approach for learning object relations, maintaining the convenience of end-to-end training while learning a structured and interpretable model of object concepts. Language-grounded learning is a hybrid approach that learns novel concept words (e.g. colors, shapes, and actions) during LfD. This approach has been used to label task components [20], learn object and action words [21], and gain multi-modal concepts via clarification dialogue [8].

ITL is another hybrid approach that fuses elements of LfD, language-grounded learning, and active learning to learn tasks as a system of rules and concepts, rather than input-output black boxes. Several works [6], [7], [22] have utilized ITL, but these have all focused on learning physical tasks. By adapting ITL for SARs, these robots will not simply mimic human behavior, but rather understand the social rules of the tasks they perform and convey their reasoning to humans.

III. Methodology

Our approach for learning social tasks via ITL[1] consists of three components: a behavior tree-based learning agent that generates dialogue, a natural language understanding (NLU) system, and a synthetic dataset for training the NLU system. When learning a new task, the learning agent prompts the teacher with questions about the task. The teacher's answers will be processed by the NLU system, which generates a sub-tree to append to the behavior tree. This process repeats until the teacher indicates teaching is complete.

A. Learning Agent

The learning agent generates dialogue with the teacher to learn a behavior tree model of the social task. A task's behavior tree can contain sequences and conditionals as interior nodes, while the leaf nodes are robot speech or listening behaviors. Sequences can be given the name of an action such as "greeting the customer". Starting with an empty sequence, the learning agent recursively searches the behavior tree for incomplete sequences or conditionals, prompting the teacher for additional information until the teacher indicates the sub-tree is complete. This process repeats until the entire tree is finalized and the task is learned. Given the goal of learning from non-experts, it cannot be assumed that teachers will be skilled in computational thinking (i.e., the ability to break a complex task down into simple

[1] https://github.com/Intelligent-Robotics-Lab/social-itl.git

components and logic). Accordingly, the learning agent uses guided prompts to induce computational thinking in the human teacher. These prompts ask for the next step while including context about the subtree that is being learned, such as the name of the sequence being learned, the current conditional statement, or the previous learned action. If the NLU system cannot understand the teacher's response, the learning agent indicates a failure and re-delivers the prompt. If the NLU system misunderstands, the teacher can indicate the misunderstanding and the learning agent will apologize and backtrack in the learning process appropriately.

B. Natural Language Understanding

The NLU system parses speech received from the teacher into a computational representation which can be used to build behavior trees. All input utterances are first classified as one of 6 possible intents: confirmation, denial, uncertainty, indication of speech misrecognition, task-relevant instructions, and completion of the task. We utilize SimCSE [23] to vectorize the input utterances and fit a weighted K-Nearest Neighbor classifier (k=5) on a set of sample utterances.

For any utterances that are classified as task-relevant instructions, the system parses a computational representation that can be returned as a sub-tree to the learning agent. We utilize a semantic parser based on a combination of the BERT [24] and T5 [25] language models. Using a similar technique to [18], a BERT model with a token classification head masks out portions of the teacher utterance referring to quotes that the robot should say or might expect a customer to say. This significantly reduces the variance in input utterances, making it easier for the parser to learn and generalize. The masked utterance is passed to a T5 sequence-to-sequence model, which converts the utterance to a computational parse. The parser is trained to parse the following constructs:

- *if(x, y)*: if x condition, do y
- *heard(x)*: return True if person says x (or something similar) to the robot
- *say(x)*: say x to the person (x is a direct quote)
- *tell(x)*: tell the person x (similar to say, but requires rephrasing to the robot perspective)
- *ask(x)*: ask x to the person
- *resolve(x)*: perform the action x

To prevent the T5 model from generating unpredictable results, the decoding vocabulary is restricted to tokens present in either the input sequence or the set of constructs listed above. Next, the masked portions of the parse are substituted with their original utterance segments to obtain a complete parse. However, some instructions contain language that must be converted to the robot's perspective for a live interaction (e.g. *tell("that they can leave their key on the desk")* should be converted to *say("you can leave your key on the desk")*). We utilize a GPT-J language model with a prompt to rephrase all utterances to the robot's perspective. After finalizing the parse, the NLU system converts the parse to a sub-tree of behaviors and returns it to the learning agent.

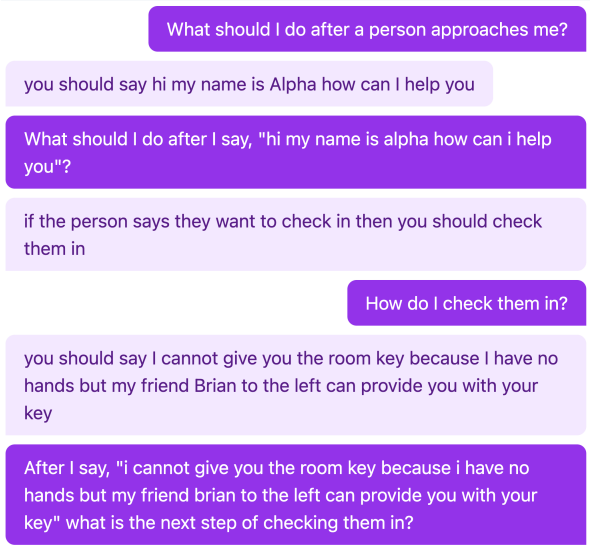

Fig. 1. Example Prompts from the learning agent: after prompt 1 a speech behavior is added, after prompt 2 a conditional is added with a new "check in" sequence as a child, and after prompt 3 the "check in" sequence is filled in with a speech behavior.

C. Synthetic Dataset

Given the difficulty of collecting and labelling a large dataset of teacher instructions and parses, we use a synthetic dataset for training the NLU system. First, we form sets of template phrases corresponding to the constructs in III-B; these can be found in the source code. We source names of actions for *resolve()* from a WikiHow dataset [26] and sample dialogue for *heard(), say(), tell(),* and *ask()* from the DailyDialog corpus [27]. Sentences and parses are generated by recursively substituting actions and dialogue into these template phrases. In total, 10,000 pairs of sentences and parses are generated for training the NLU system.

D. Task Execution

To perform the task after teaching, the robot ticks through the behavior tree. When arriving at a *heard(x)* behavior, the robot listens for a customer response and utilizes the sentence vector cosine distance to determine if the distance from phrase x is > 0.4 (an empirically determined threshold), and returns *Success* if true and *Failure* otherwise. If *heard(x)* fails, subsequent *heard(x)* behaviors will not stop to listen to the customer until either a *heard(x)* behavior returns Success or an *else* statement is reached. This design enables a fallback flow where a single robot listen can be matched against multiple phrases $x_1, x_2, ..., x_n$ When the end of the behavior tree is reached, execution repeats from the beginning.

IV. Experiments

A. Study Design

To evaluate the performance of our system, we designed an HRI study where participants teach a Furhat social robot (named Alpha) to be a hotel concierge. We utilize a within-subjects design where participants taught the hotel concierge task to the robot with the proposed ITL system (Figure 3) and again with LfD. The presentation of conditions was balanced.

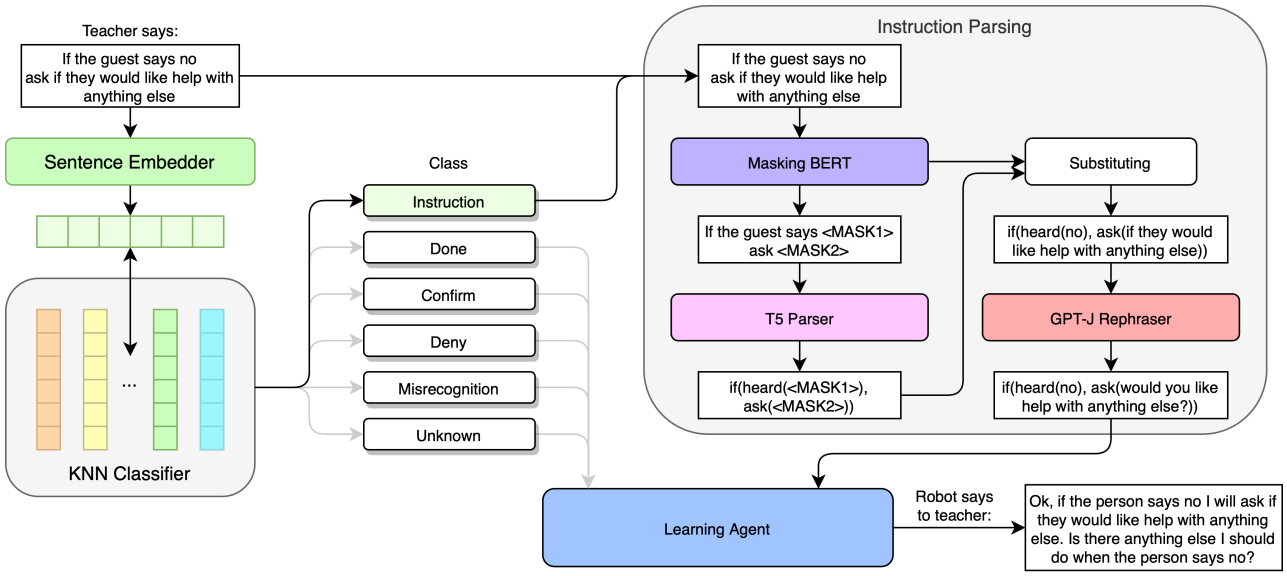

Fig. 2. Instruction parsing system with learning agent

1) Participants: We recruited 16 native English speakers as study participants. Two participants were excluded due to system errors. Of the remaining 14, there were 6 females and 8 males, with a mean age of 32.1 years ($\sigma = 15.56$).

2) Procedures: Before beginning LfD or ITL, a researcher demonstrated how to teach the robot a separate sub-task (assisting the guest with towels), which included showing how to correct a misunderstanding. Participants could request that the demonstration be repeated at any point in the experiment. No further guidance was given for teaching the robot with ITL, as we wanted to investigate the intuitiveness of this learning system without external help. Participants were then provided with a paper task description for a hotel concierage job, which consisted of 6 main sub-tasks: greeting the guest, checking in the guest, assisting the guest with luggage, checking out the guest, providing information on hotel amenities, and providing information on local restaurants.

The participants were then asked to teach the robot this task. While teaching under both conditions, participants had access to the task description and a touchscreen displaying the conversation history, enabling them to better detect automatic speech recognition (ASR) errors and review what was already taught. Participants were instructed that the robot should perform these sub-tasks based on the hotel guest's needs and not simply one after the other. This provides a hint to include conditional logic when teaching the robot. After teaching, they played a hotel guest to assess the performance of the robot. Based upon their impression of the robot's performance they could choose to re-teach the robot. In the ITL condition, re-teaching involved starting from the beginning, while in the LfD condition it consisted of providing more demonstrations to the existing model.

B. LfD System

In the LfD scenario, the participant played the role of the robot while a researcher acted as the hotel guest. The concierge task was then taught by mock dialogue between them. The participant was responsible for designing both the hotel guest and concierge script to avoid the researcher biasing the dialogue. Participants could practice demonstrations with the researcher before recording data for LfD. They could test the LfD model by acting as a customer and interacting with the robot concierge running the model. This allowed them to identify undemonstrated states or corrupted actions and provide more demonstrations as needed. The robot's included microphone array allowed for separating the dialogue of the two speakers; this worked well but occasionally matched speech to the wrong speaker, especially short responses. To learn a policy, we utilize a similar approach as [11] and [1], but our method is designed for one-shot learning from demonstration so that it can be completed in the same amount of time it takes to teach with ITL ($\sim$15 minutes for the six sub-tasks). This approach omits the clustering used in the prior approaches and uses a nearest-neighbor approach to select the current robot action a_t based upon the last robot action a_{t-1} and the guest's response to it, s_t. We define the distance d between such (a, s) pairs as:

$$\begin{aligned} d((a_1, s_1), (a_2, s_2)) = 0.2 * \|v(a_1) - v(a_2)\| \\ +0.8 * \|v(s_1) - v(s_2)\| \end{aligned} \quad (1)$$

where $v()$ denotes the sentence vector computed by SimCSE.

C. Evaluation Procedure

In this experiment we evaluate the performance of both learning models and compare participants' trust in the robot and perceived workload between the ITL and LfD conditions.

1) Performance Evaluation: To measure performance of the models, each participant (except the final participant) played a customer while interacting with the LfD and ITL models trained by the previous participant. Two human

Fig. 3. Teaching the Furhat robot with ITL

coders labeled each robot action as appropriate or inappropriate based on the customer's responses. They also labeled the category of action the robot should perform, either one of the six sub-tasks or an "other" category for general dialogue such as "you're welcome". Participants and coders were unaware which teaching method was used to train the model. Actions where the robot repeats ASR errors from the training procedure (e.g. *would you like to check it* vs. *would you like to check in*) were coded as "appropriate w/ ASR error" because they could be eliminated with improved ASR. Sections with low agreement were cooperatively re-coded. The final Cohen's kappa agreement for was 0.91 for category and 0.90 for action appropriateness.

2) Participant Attitudes: We investigated participants' trust towards the robot and perceived workload in both teaching scenarios. Trust was measured using the abbreviated 14-item version of the Trust Perception Scale-HRI questionnaire [28]. Perceived workload was measured using the NASA-TLX scale [29]. Both questionnaires were administered immediately after participants taught and tested their own model under the respective LfD/ITL condition, but before evaluating the previous participant's model. We also asked participants to select which teaching style they preferred and describe their reasons why. Finally, we asked them to rank their computer programming experience on a 5 point scale.

V. Results

A. Questionnaire

The results of our HRI questionnaires are illustrated in Table I and Figure 4. Trust in the LfD condition (70.4%) was higher than in the ITL condition (64.3%), but using a paired t-test we found this effect was not significant ($t(13)$=1.70, p=0.11). Perceived workload was non-normal, so we utilized a Wilcoxon signed-rank test to analyze the difference. Workload was higher under the ITL condition (55.5%) than the LfD condition (46.8%), but this effect was also not significant (Z=-1.02, p=0.15). More participants indicated a preference for LfD (9) than ITL (5). To analyze whether computational thinking correlated with these metrics, we compared the relative difference of trust and workload between the two conditions against participants' self evaluated programming experience using a Spearman correlation test. There was no correlation between programming experience and trust differences (p=0.38) or workload differences (p=0.45).

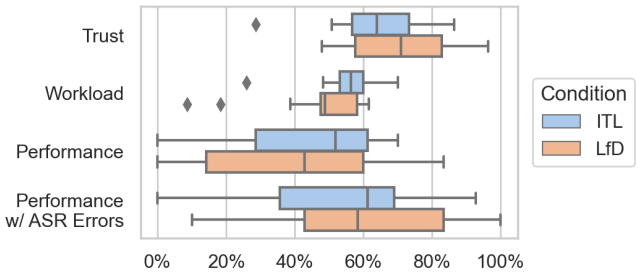

Fig. 4. Distributions of Trust, Workload, and Total Performance

TABLE I
Questionnaire Results

	Trust %		Workload %		Preferred Teaching Method
	μ	σ	μ	σ	
LfD	70.4	14.9	46.8	15.1	9
ITL	64.3	14.7	55.5	9.9	5

B. Performance Results

The results of our performance evaluation are shown in Table II and Figure 4. For each participant, we compute the percentage of appropriate actions, including and excluding ASR errors, for each of the 7 action categories. The category scores of each coder are averaged for each participant. A participant's total score is defined as the mean of these 7 categories. The categorical and total scores in Table II represent the mean of each participant's categorical and total scores. This mean-of-means design ensures even weighting among participants with varying amounts of evaluation dialogue.

In total, our performance data consists of 372 actions. ITL performance (47.6%) is higher than LfD (39.3%), but when disregarding ASR errors the LfD system slightly outperforms the ITL system (60.8% vs. 54.5%). However, neither of these differences were found to be significant. As shown in Figure 4, there is high variance with both teaching styles, with some teachers achieving near-perfect performance while others had zero performance. Spearman tests did not show correlation between programming experience and performance (ignoring ASR errors) with ITL (p=0.84) or LfD (p=0.72).

TABLE II
Performance Results

	% of Appropriate Actions		% of Appropriate Actions ignoring ASR Errors	
	ITL	LfD	ITL	LfD
Greeting	72.9	60.0	72.3	66.2
Check In	51.0	34.6	57.3	69.6
Luggage	38.5	45.0	38.5	54.0
Check Out	45.8	51.8	46.3	56.9
Amenities	25.0	4.2	44.4	62.5
Restaurants	33.3	55.1	54.2	82.1
Other	67.3	9.5	67.3	9.5
Total	47.6	39.3	54.5	60.8

The differing performance across sub-tasks reveals some of the relative strengths of ITL and LfD. Amenities is a task with complex vocabulary: this task illustrates how ITL enables better correction of ASR errors. Meanwhile, simple tasks like Greeting (where ASR errors are uncommon) show less difference between ITL and LfD. Sub-tasks appear in Table II in the same order as the task description provided to participants, and many participants chose to teach in this order. The decreased accuracy for later tasks such as Amenities and Restaurants illustrates the problem of unintended temporal dependency, discussed in section V-D.2.

C. Participant Feedback

As shown in Table I, participants mainly preferred LfD over ITL. In the free-response questions, those favoring LfD largely cited ease of use (n=8) and better evaluation performance (n=5) as their reasons. Of those favoring ITL, most indicated it provided greater learning transparency (n=3) and enabled improved performance by correcting ASR errors (n=3). Individuals from both groups said that ITL had a higher learning curve (n=9), but for some the potential for increased performance outweighed this. As one participant shared, "[with ITL] I feel like even though I was unsuccessful in training the robot, it would be more likely to perform appropriately when trained successfully."

D. Teaching Analysis

We reviewed the transcripts of participants teaching the robot to identify successes and challenges in each condition.

1) Successful Teaching: With LfD, the most successful teachers demonstrated a reasonable range of hotel guest intents and responses. These teachers avoided ASR errors by speaking clearly and pausing between dialogue turns. They also utilized the tablet interface to identify ASR errors immediately and correct them with more demonstrations.

With ITL, successful teachers used conditional statements well to model different conversational branches. They understood when a conditional should end, meaning the underlying behavior tree was wide and only had nested behaviors where necessary. They also developed a model of what phrases the robot could and could not understand (sometimes remarking out loud) and phrased commands accordingly. If the robot did not understand, the teacher explored different phrasing styles. Patience also contributed to successful teaching: several participants had low performance on their first attempt but significantly improved with another teaching session.

2) Failure Modes: When participants struggled to teach the robot, we identified the following patterns:

Technical Challenges: A common challenge was uncorrected ASR errors. In both conditions the robot sometimes heard incorrectly, but in LfD the robot could also assign utterances to the wrong person. Teachers corrected ASR performance more often with ITL than LfD, likely because misunderstandings were more immediately apparent in ITL as the robot would always repeat back its understanding of the teacher's instructions. While some participants achieved better performance through ITL by resolving ASR errors, participants who focused too much on ASR errors could get stuck in failure loops. The robot could not understand some phrases even with perfect enunciation, but teachers made repeated attempts (as many as 7) to notify the robot it misunderstood and retry, rather than simply continuing with teaching. Teachers also tried to tell the robot to replace an individual word, but the learning agent could only replace full utterances. One participant suggested typing as a much less frustrating alternative. Such failure loops seemed to increase frustration and wasted time that could otherwise be spent improving the robot's task model.

Computational Thinking Challenges: Several participants failed to teach the robot to respond dynamically, so the robot simply listed off information without first listening for a customer's needs. This failure occurred in both LfD and ITL, but more frequently in ITL. One common difficulty with LfD was forgetting about undemonstrated states. For example, many participants started by asking "would you like to check in?" and acting out the scenario where the hotel guest said yes, but forgot to demonstrate a scenario where the guest said no. Such difficulties were not present under ITL, as the robot explicitly asks about else conditions.

Mental Model Mismatches: The most common struggle with ITL was failure to understand the temporal constraints of the behavior tree that was being generated. Despite the design of the learning agent prompts to induce computational thinking, many teachers created undesirable temporal dependencies. For example, the robot might only provide information on hotel amenities immediately after a guest asks about check in; if the guest does not ask about check in that part of the behavior tree remains inaccessible. In several participants this was so prevalent that the robot could only perform the initial behavior (check in) successfully. Another type of failure was when participants over-simplified the task to an extent the ITL algorithm could not understand. For example, "'What should I do next?', 'Listen for a response', 'How do I listen for a response?', 'Wait for the guest to say something', 'How do I wait for the guest to say something?', ..." In some cases, the participant was frustrated enough to give up before teaching all 6 sub-tasks, causing low performance. Some teachers also tried to teach slot-filling behaviors to the robot, such as asking for the guest's name and reusing that information later in the dialogue, but currently the ITL system does not support this.

VI. Discussion

In this work, we present a novel approach to learn social interactions via interactive task learning and conduct an exploratory study to compare the performance and teacher perceptions of the system against an LfD approach. We did not find significant differences in workload, trust, or performance between the two systems, but this experiment is limited by the small sample size and large variance. Nonetheless, the HRI study performed in this work sheds light on how humans attempt to teach social tasks via ITL and areas for improvement. We did not observe correlations between computer programming experience and trust, workload, or

performance in either condition, which could suggest computational thinking is not the main obstacle for ITL teaching. Rather, faulty mental models of the robot's knowledge may be limiting for programmers and non-programmers alike.

From an algorithmic standpoint, the NLU system can be enhanced to improve the robot's capabilities and reduce teaching difficulties. Although the study focused on simple static behaviors, real-world interactions require a robot to store information (such as names) and adapt phrases accordingly. We leave such learnable slot-filling NLU features for future work. Similar frustrations with ASR have been observed in conversational computer programming systems [30]. Enabling word-level instead of sentence-level ASR correction could reduce frustration, save time, and enhance system performance. Moreover, training the NLU system to detect faulty mental models would allow the learning agent to address misunderstandings, preventing a failure loop.

Additional methods for improving teachers' mental model could also be investigated. The common issue of unintentional temporal dependencies could likely be resolved by better prompting. For example, adding a prompt such as *"Should I do behavior x only as part of behavior y, or any time z happens?"* could significantly reduce such failures. Utilizing visual aids beyond a basic transcript, such as a simplified abstraction of the model/behavior tree, could also aid teachers in understanding the robot's task model [30].

Finally, a hybrid approach involving LfD and ITL could combine the best parts of both methods. LfD could present a simple way of initially teaching, and ITL could be used to clarify temporal dependencies and fix ASR errors. Humans naturally utilize such a multi-modal teaching approach with each other, which could make a hybrid learning approach a more natural and intuitive way to teach social robots.

VII. Acknowledgements

We would like to thank Evan Dallas and Iman Bakhoda for coding the data from this experiment.

References

[1] A. Tyshka and W.-Y. G. Louie, "Transparent learning from demonstration for robot-mediated therapy," in *Proc. 31st IEEE Int. Conf. Robot and Human Interactive Communication*. IEEE, 2022, pp. 891–897.

[2] W. Y. G. Louie and G. Nejat, "A social robot learning to facilitate an assistive group-based activity from non-expert caregivers," *Int. Journal of Social Robotics*, vol. 12, pp. 1159–1176, Nov. 2020.

[3] S. Thellman and T. Ziemke, "The Perceptual Belief Problem," *ACM Trans. on Human-Robot Interaction*, vol. 10, no. 3, pp. 1–15, Jul 2021.

[4] J. Y. Chai, Q. Gao, L. She, S. Yang, S. Saba-Sadiya, and G. Xu, "Language to action: Towards interactive task learning with physical agents," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 2–9.

[5] J. E. Laird, *et al.*, "Interactive task learning," *IEEE Intelligent Systems*, vol. 32, no. 4, pp. 6–21, 2017.

[6] J. R. Kirk and J. E. Laird, "Learning hierarchical symbolic representations to support interactive task learning and knowledge transfer," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 6095–6102.

[7] G. Suddrey, B. Talbot, and F. Maire, "Learning and executing re-usable behaviour trees from natural language instruction," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 643–10 650, 2022.

[8] J. Thomason, *et al.*, "Improving grounded natural language understanding through human-robot dialog," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2019, pp. 6934–6941.

[9] P. Ramaraj, C. L. Ortiz, and S. Mohan, "Unpacking human teachers' intentions for natural interactive task learning," in *Proc. 30th IEEE Int. Conf. Robot and Human Interactive Commun.*, 2021, pp. 1173–1180.

[10] P. Ramaraj, S. Sahay, S. H. Kumar, W. S. Lasecki, and J. E. Laird, "Towards using transparency mechanisms to build better mental models," in *Advances in Cognitive Systems: 7th Goal Reasoning Workshop*, vol. 7, 2019, pp. 1–6.

[11] P. Liu, D. F. Glas, T. Kanda, and H. Ishiguro, "Data-driven hri: Learning social behaviors by example from human–human interaction," *IEEE Transactions on Robotics*, vol. 32, no. 4, pp. 988–1008, 2016.

[12] S. Stepputtis, J. Campbell, M. Phielipp, S. Lee, C. Baral, and H. Ben Amor, "Language-conditioned imitation learning for robot manipulation tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 139–13 150, 2020.

[13] C. Lynch and P. Sermanet, "Language Conditioned Imitation Learning over Unstructured Data," in *Robotics: Science and Systems*. MIT Press Journals, May 2021.

[14] G. Papagni and S. Koeszegi, "Understandable and trustworthy explainable robots: A sensemaking perspective," *Paladyn, Journal of Behavioral Robotics*, vol. 12, no. 1, pp. 13–30, 2021.

[15] K. French, S. Wu, T. Pan, Z. Zhou, and O. C. Jenkins, "Learning behavior trees from demonstration," in *2019 International Conference on Robotics and Automation*, 2019, pp. 7791–7797.

[16] Y. Hristov, A. Lascarides, and S. Ramamoorthy, "Interpretable latent spaces for learning from demonstration," in *Conference on Robot Learning*. PMLR, 2018, pp. 957–968.

[17] D. Zhang, Q. Li, Y. Zheng, L. Wei, D. Zhang, and Z. Zhang, "Explainable Hierarchical Imitation Learning for Robotic Drink Pouring," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 3871–3887, 2021.

[18] N. Walker, Y.-T. Peng, and M. Cakmak, "Neural Semantic Parsing with Anonymization for Command Understanding in General-Purpose Service Robots," *Lecture Notes in Computer Science*, vol. 11531 LNAI, pp. 337–350, Jul 2019.

[19] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu, "The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision," in *Proceedings of the International Conference on Learning Representations*, 2019.

[20] C. Liu, *et al.*, "Jointly learning grounded task structures from language instruction and visual demonstration," in *Proc. Conf. Empirical Methods in Natural Language Processing*, 2016, pp. 1482–1492.

[21] M. Hirschmanner, S. Gross, S. Zafari, B. Krenn, F. Neubarth, and M. Vincze, "Investigating transparency methods in a robot word-learning system and their effects on human teaching behaviors," *30th IEEE Int. Conf. Robot Human Interactive Commun.*, pp. 175–182, 2021.

[22] A. Mohseni-Kabir, C. Rich, S. Chernova, C. L. Sidner, and D. Miller, "Interactive hierarchical task learning from a single demonstration," in *Proc. 10th Annu. ACM/IEEE Int. Conf. on Human-Robot Interaction*, 2015, pp. 205–212.

[23] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *Proc. of the Conference on Empirical Methods in Natural Language Processing*, Nov. 2021, pp. 6894–6910.

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[25] C. Raffel, *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[26] S. Zhou, *et al.*, "Show me more details: Discovering hierarchies of procedures from semi-structured web data," in *Annual Conference of the Association for Computational Linguistics*, May 2022.

[27] Y. Li, H. Su, X. Shen, W. Li, Z. Cao, and S. Niu, "DailyDialog: A manually labelled multi-turn dialogue dataset," in *Proc. 8th Int. Joint Conf. Natural Lang. Process.*, vol. 1, Nov. 2017, pp. 986–995.

[28] K. E. Schaefer, "Measuring trust in human robot interactions: Development of the "trust perception scale-hri"," in *Robust Intelligence and Trust in Autonomous Systems*. Boston, MA: Springer, 2016, pp. 191–218.

[29] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," in *Advances in psychology*. Elsevier, 1988, vol. 52, pp. 139–183.

[30] J. Van Brummelen, K. Weng, P. Lin, and C. Yeo, "Convo: What does conversational programming need?" in *2020 IEEE Symp. on Visual Languages and Human-Centric Computing*, 2020, pp. 1–5.