Interactive Task Learning for Social Robots: A Pilot Study

Alex 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 can 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, current research is focusing on how to teach robots specific tasks such as performing household chores [1], leading group activities [2], or delivering therapeutic interventions [3]. However, the number of social tasks for robots to perform is far beyond the range of tasks that can be trained in a lab by experts. For SARs to adapt to a wide range of applications, it is imperative that non-expert users can teach and adapt robots in the wild.

A popular approach for this is learning from demonstration (LfD), where a human demonstrates how to perform a task and the robot forms a model that can be 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 [4] 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 [5], [6]. Perceptual belief can significantly impair human robot teaching because if the teacher does not understand what concepts the robot already knows and what it needs to learn, the teacher cannot teach the robot effectively. For SARs to achieve a higher level of autonomy, it is necessary that they be able to rapidly acquire new concepts and be able to convey the extent of their knowledge to their teachers.

Interactive task learning (ITL) is a new learning paradigm that seeks to address this problem [7]. ITL expands LfD to include natural language as well as demonstration, with the aim of mimicking human learning and more closely integrating the teacher. We rarely utilize demonstration only when teaching other humans. Rather, we verbally describe the task and supplement this with demonstration. 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 [8].

Recent works have predominantly explored the use of ITL in physical manipulation tasks [7], [9]–[12] but this approach has significant potential to address open challenges in social robotics. Data-efficiency is one such challenge for SARs because while physical manipulation robots can easily gather data in simulation, 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 either crafting feature sets ahead of time (reducing generalization capability) or learning from raw data using deep neural networks (DNNs), which can struggle on limited training data. Using natural language to teach social robots has the potential to significantly reduce the amount of demonstration data needed because concepts, rules, and constraints of the task can be directly described rather than implicitly inferred from patterns in a large dataset.

However, existing approaches to ITL designed for teaching robot-object interaction are not readily applicable for teaching social human-robot interactions. In a physical 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 on objects through gestures [13] or physically demonstrating actions. In most physical manipulation experiments (e.g. [9], [13]), the environment has a finite set of objects present in the

*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)

experiment or simulation, which has the practical effect of constraining the vocabulary used. In contrast to physical manipulation scenarios, social tasks are more abstract in nature and difficult to ground. States and actions are based on the dialogue and not the physical environment. This means that concepts must be learned verbally and without the aid of physical and/or environmental teaching cues. Additionally, without a physical environment constraining the vocabulary, a teacher could say almost anything when teaching a social robot. To provide a reasonable response, a social ITL system must have a robust parsing system 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 for humans to teach to robots than physical tasks. While physical tasks or games are often intuitive to break down into rules, steps, and subtasks, social tasks can be much less structured and rely on human intuition as well as common sense. If ITL is to be used for these tasks, it must 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 utilize and perceive existing ITL systems. Questions have yet to be answered on how intuitive the teaching process is, what mental models teachers form about the robot, and whether they prefer this learning paradigm over other techniques like LfD. Some research has investigated human teaching behavior in Wizard-of-Oz studies [14] as well as virtual studies identifying failure cases [15], 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 real-world use cases.

In this work, we present a preliminary algorithmic approach for a SAR to learn social tasks via ITL and evaluate user perceptions of this system with an HRI study. This approach leverages recent advances in natural language processing to adapt to a wide range of unstructured language without the need for extensive handcrafted rules. A learning agent guides the human teacher through the process of ITL, while attempting to induce computational thinking. We share the source code of this system for reproducibility and future work[1]. 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 novel social interactions to robots [2], [3], [16]. However, demonstration can take significant time to teach tasks to robots, 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 [17], [18], 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 to new tasks. However, interpretability is especially important for social robots, where inappropriate behavior can be particularly detrimental. Global interpretability can help teachers understand what has been learned, and local interpretability can improve accuracy and user trust by rationalizing individual decisions [19]. While a number of works [20], [21] have explored interpretability techniques for LfD, a fundamental conflict arises between generalization and explanation quality. High-level, natural language explanations, as used in [1], are the kind most suitable for non-expert users. However, these approaches are frequently generated by end-to-end neural networks, meaning they require thousands of labeled explanations of a task. The end-to-end design also prevents knowledge from being transferred from one model to another, as there is no explicit modeling of concepts, only a latent space. Additionally, because these explanations are labeled after the robot has learned the task, such methods cannot provide interpretability while a person is teaching the robot. Alternative approaches [20] leverage inherently-interpretable models, but these cannot explain in natural language and high-level concepts, and therefore are more oriented towards model transparency for expert users [22].

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. [23] present an approach to language-conditioned learning that parses an intermediate representation in logical grammar, which provides an interpretable model and enables generalization to some unseen tasks. A neuro-symbolic approach to learning object relations is presented in [24]. This hybrid design maintains the convenience of end-to-end training while learning a structured and interpretable model of object concepts. Language-grounded learning is another hybrid approach that seeks to learn discrete concepts (e.g. colors, shapes, and actions) during the process of learning from demonstration. Instead of relying on an extensive pre-existing knowledge base, robots can learn common sense knowledge in-situ, improving generalization to new tasks and reducing initial development effort. Language-grounded learning has been utilized to provide labels for components of a task [25], learn object and action words [26], and learn multi-modal concepts through clarification dialogue [13].

Another hybrid approach, interactive task learning [8], aims to learn tasks as a systematic hierarchy of rules and

[1]https://github.com/Intelligent-Robotics-Lab/multimodal-learning.git

concepts, rather than simply input-output black boxes. Where LfD only observes *what* a person did, ITL intends to also extract *why* the person performed that action. By learning rules explicitly rather than inferring them from patterns, ITL aims to learn tasks faster, with less susceptibility to noise, and with greater transparency. ITL accomplishes this by fusing LfD, language-grounded learning, and active learning. Several works [9]–[12] have utilized ITL, but these have all focused on learning physical tasks and concepts. 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 back to humans.

III. Methodology

Our approach for learning social tasks via ITL 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 to these questions 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 to interact with the human teacher and learn a behavior tree model of the social task. The behavior tree for a task can contain sequences and conditionals as interior nodes of the tree, while the leaf nodes can consist of robot speech or listening behaviors. Sequences can be associated with the name of an action such as "greeting the customer". The behavior tree starts with an initially empty sequence. The learning agent performs a recursive search on the behavior tree for sequences or conditionals that have not been finalized, and prompts the teacher for further information until the teacher indicates this sub-tree is finalized. This process repeats until the entire tree is finalized and the robot is ready to perform the task. 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 components and logic). Accordingly, the learning agent uses guided prompts to induce computational thinking in the human teacher. These prompts include context about the subtree that is currently being learned, such the name of the sequence being learned, the current conditional statement, or the previous learned action, and ask for the next step. If the NLU system is unable to understand the teacher's response, the learning agent indicates the failure and re-delivers the prompt. If the NLU system misunderstands the teacher, 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 is responsible for parsing speech received from the teacher into a computational representation the learning agent can use to build behavior trees. All input utterances are first classified based on the intent. We consider 6 possible intents: confirmation, denial, uncertainty, indication of speech misrecognition, task-relevant instructions, and completion of the task. We utilize SimCSE [27] 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. In order to do this, we utilize a semantic parser based on a combination of the BERT and T5 language models. Inspired by the anonymization technique from [23], 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 technique significantly reduces the variance in input utterances, making it easier for the parser to learn and generalize. The masked utterance is then fed 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 mitigate the potential for the T5 model to generate 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 can 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 model with a prompt to rephrase all utterances to the robot's perspective. Once the parse has been finalized, the NLU system converts the parse text to a sub-tree of behaviors that is returned to the Learning Agent.

C. Synthetic Dataset

Given the difficulty of collecting and labelling a large dataset for mapping teacher instructions to parses, we utilize a synthetic dataset for training the NLU system. To generate such data, we form sets of primitive phrases corresponding to the constructs in III-B. We source names of actions for *resolve()* from a WikiHow dataset and sample dialogue for *heard(), say(), tell(),* and *ask()* from the DailyDialog corpus. Full sentences and parses are generated by recursively substituting these primitive phrases. The full list of primitive phrases can be found in the source code. In total, 10,000 pairs of sentences and parses are generated for training the NLU system.

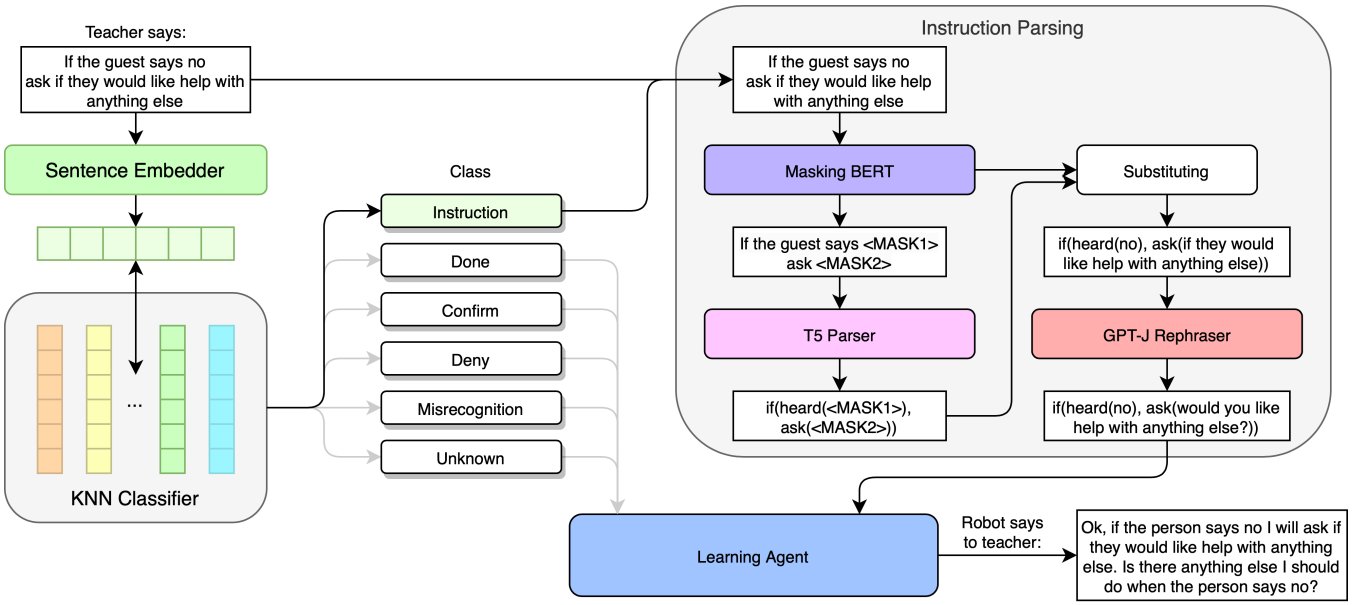

Fig. 1. Instruction parsing system

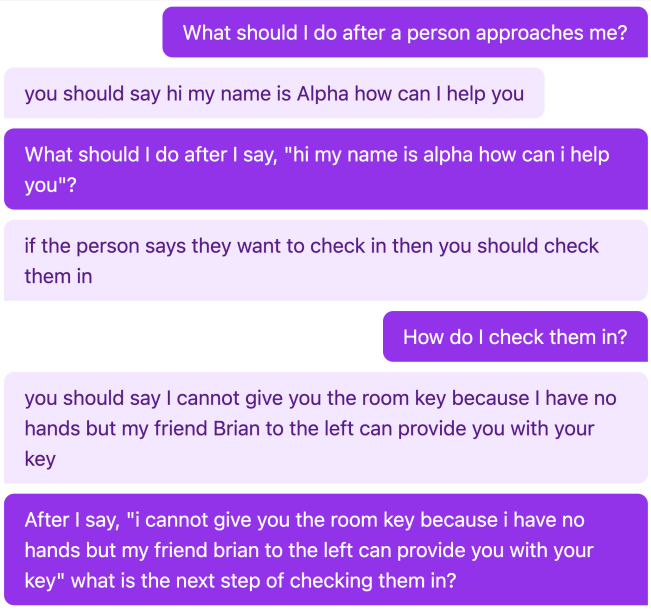

Fig. 2. Example Prompts from the Learning Agent: after prompt 1 a speech behavior is added, after prompt 2 a conditional is added with a new "check in" sequence as a child, after prompt 3 the "check in" sequence is filled in with a speech behavior.

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 $x_1, x_2, etc.$ When the end of the behavior tree is reached, execution repeats from the beginning.

Fig. 3. Teaching the Furhat robot with ITL

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.

1) Participants: We recruited 16 native English speakers as participants for our study. 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 brief demonstration was provided where the researcher taught the robot a separate sub-task (assisting the guest with towels). This demonstration included showing how to correct a misunderstanding. Participants could request that the demonstra-

tion be repeated at any point in the experiment. Participants received no further guidance on how to teach the robot with ITL, as we wanted to investigate how intuitive this learning system is without external help. Participants were then provided with a 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
- Providing information on local restaurants

The participants were then asked to teach the robot this task. During the teaching process for both conditions, participants had access to a touchscreen displaying the conversation history, enabling them to better detect automatic speech recognition (ASR) errors and review what was already taught. While teaching under both conditions, 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 teachers that they should include conditional logic when teaching the robot. After teaching, they played a hotel guest to assess the performance of the robot and 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 re-teaching 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 had the opportunity to act out mock conversations with the researcher before recording data for LfD. Participants were able to test the LfD model by acting as the customer and interacting with the robot concierge running the LfD policy. This allowed them to identify undemonstrated states or corrupted actions and provide more demonstrations as needed. The microphone array included with the robot allowed for separating the dialogue of the two speakers; this worked well but occasionally attributed utterances to the wrong speaker, especially short responses. For learning a policy, we utilize a similar approach as [3], [16], 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 one-shot approach omits the clustering used in the previous 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:

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

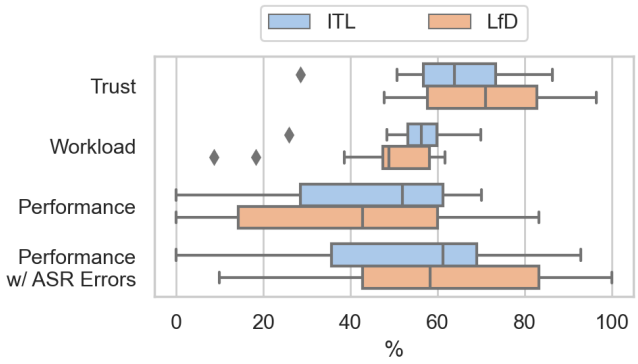

Fig. 4. Distributions of Trust, Workload, and Total Performance

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

C. Evaluation Procedure

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

1) Performance Evaluation: To quantitatively measure performance of the models, we had participants play a customer while interacting with the robot in the hotel scenario. Participants tested the LfD and ITL models that the previous participant trained. The final participant's trained models were not evaluated. Two human coders labeled each robot action as either appropriate or inappropriate based on the customer's responses. Additionally, the coders 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". Neither the participant nor coder knew whether LfD or ITL was used to train the model. Robot actions where the robot repeats back 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: In addition to evaluating the model performance, 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 the participant taught the robot and tested their own model under the respective LfD/ITL condition, but before evaluating the previous participant's models. We also directly asked participants to select which teaching style they preferred and describe their reasons why. Finally, we asked participants to rank their computer programming experience on a 5 point scale.

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

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

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 found to be 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 participant's 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).

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 and appropriate actions ignoring ASR errors for each of the 7 action categories. The category scores of each coder are averaged for each participant. A participant's total performance score (as shown in Figure 4) 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 (4.76%) is higher the LfD performance (39.3%). However, disregarding ASR errors the LfD system slightly outperforms the ITL system (60.8% vs. 54.5%). Amenities is a task with complex vocabulary, this category illustrates how ITL provides greater ability to correct ASR errors. Meanwhile, simple tasks such as Greeting (where ASR errors are uncommon) do not show as much difference between the two teaching methods. Sub-tasks appear in Table II in the same order as the task description provided to participants, and many participants chose to teach the tasks 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. As shown in Figure 4, there is significant variance with both teaching styles, with some participants achieving near-perfect performance while others had zero performance. Again, we ran Spearman tests to determine whether programming experience correlated with ITl or LfD performance (ignoring ASR errors). Correlation was not observed with ITL (p=0.84) or LfD (p=0.72) performance.

C. Participant Feedback

As indicated in Table I, participants predominantly preferred LfD over ITL. Analyzing the open-ended questions, those favoring LfD largely cited ease of use (n=8) and better post-teaching task performance (n=5) as their reasons. Among those who favored ITL, most indicated that ITL provided greater transparency in what was learned (n=3) and enabled them to improve 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 designed conversations that reflected a reasonable range of hotel guest intents and responses. These teachers were careful to avoid ASR errors by speaking clearly and leaving pauses between conversational turns. They also utilized the tablet interface to identify ASR errors immediately and provide more demonstrations to correct them.

With ITL, successful teachers used conditional statements well to model different conversational branches. They correctly followed the robot's prompts to understand when a conditional statement 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 failed to understand, the teacher explored different phrasing styles rather than persisting with the same phrase. Patience

also contributed to successful teaching: several participants did not achieve optimal 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 there was the additional difficulty of the robot assigning 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, if participants focused too much on ASR errors it could also lead to participants getting stuck in failure loops. Some phrases were simply unable to be understood correctly by the robot 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 tried to also tell the robot to replace an individual word, (e.g., "Replace robot lead exercise with robot led exercise") 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 based upon the customer's needs, so the robot simply listed off information without first identifying a customer's needs by listening. 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 accidentally created temporal dependencies where there should not have been. For example, when taught this way the robot would 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 loop occurred 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 frustration was high enough for the participant to give up on teaching before all 6 sub-tasks had been taught, causing low performance. Some teachers also attempted 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 learning agent and NLU were only capable of rote memorization.

VI. Discussion

In this work, we present a novel approach to learning 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. Overall, the strengths of the ITL system include better performance without ASR errors and greater transparency in what was learned. This result is promising given the inevitability of ASR errors with current technology. Meanwhile, the simpler LfD system was considered easier to use by participants and outperformed ITL when disregarding ASR errors. We also did not observe correlations between computer programming experience and participant attitudes or performance in either condition. This finding could suggest that computational thinking is not the main obstacle for effective ITL teaching. Rather, faulty mental models of the robot's knowledge may be a limiting factor for programmers and non-programmers alike.

The HRI study performed in this work sheds light on how humans attempt to teach social tasks via ITL and areas for improvement. From an algorithmic standpoint, the NLU system could be improved in several ways to make the robot more capable and reduce teaching difficulties. The task scenario was simplified to static behaviors for this study, but for real-world interactions a robot must be able to store information from an interaction, such as names, and adapt phrases accordingly. We leave such learnable slot-filling NLU features for future work. Features such as word-level (rather than sentence-level) ASR correction could also reduce frustration and time spent while increasing system performance. Similar sentiments pertaining to ASR frustrations have been observed with ASR-based conversational computer programming systems [30]. Additionally, if the NLU system were trained to detect mismatches in the mental model, these miscommunications could be addressed by the learning agent rather than triggering a failure loop.

However, other improvements could be investigated for improving teachers' mental model of the robot. The common issue of unintentional temporal dependencies could likely be resolved by better prompting language. For example, adding *"Should I do behavior x only as part of behavior y, or any time z happens?"* to the learning agent prompts might eliminate many occurrences of this failure. The use of a visual aid beyond the simple tablet transcript provided in this study could also assist the participant in understanding the robot's model of the task [30]. This could be an illustration of the model/behavior tree itself or a simpler interface. Regardless, designing it for those without computer science experience would be essential [31].

Finally, a hybrid approach involving LfD and ITL could combine the best qualities 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] 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*, 2021.

[2] W. Y. G. Louie and G. Nejat, "A social robot learning to facilitate an assistive group-based activity from non-expert caregivers," *International Journal of Social Robotics*, vol. 12, pp. 1159–1176, 11 2020.

[3] A. Tyshka and W.-Y. G. Louie, "Transparent learning from demonstration for robot-mediated therapy," in *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2022, pp. 891–897.

[4] A. Hijaz, J. Korneder, and W. Y. G. Louie, "In-the-wild learning from demonstration for therapies for autism spectrum disorder," *2021 30th IEEE International Conference on Robot and Human Interactive Communication, RO-MAN 2021*, pp. 1224–1229, 2021.

[5] S. Thellman and T. Ziemke, "The Perceptual Belief Problem," *ACM Transactions on Human-Robot Interaction*, vol. 10, no. 3, Jul 2021.

[6] M. Kwon, M. F. Jung, and R. A. Knepper, "Human expectations of social robots," in *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2016, pp. 463–464.

[7] 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 *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, 7 2018, pp. 2–9.

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

[9] J. R. Kirk and J. E. Laird, "Learning Hierarchical Symbolic Representations to Support Interactive Task Learning and Knowledge Transfer," 2019.

[10] A. Mohseni-Kabir, C. Rich, S. Chernova, C. L. Sidner, and D. Miller, "Interactive hierarchical task learning from a single demonstration," in *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, 2015, pp. 205–212.

[11] 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.

[12] P. Lindes, A. Mininger, J. R. Kirk, and J. E. Laird, "Grounding language for interactive task learning," in *Proceedings of the First Workshop on Language Grounding for Robotics*, 2017, pp. 1–9.

[13] J. Thomason, *et al.*, "Improving grounded natural language understanding through human-robot dialog," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, pp. 6934–6941, May 2019.

[14] P. Ramaraj, C. L. Ortiz, and S. Mohan, "Unpacking human teachers' intentions for natural interactive task learning," in *2021 30th IEEE International Conference on Robot & Human Interactive Communication*. IEEE, 2021, pp. 1173–1180.

[15] 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.

[16] 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.

[17] 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.

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

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

[20] 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 (ICRA)*, 2019, pp. 7791–7797.

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

[22] U. Bhatt, *et al.*, "Explainable machine learning in deployment," in *Proceedings of the 2020 conference on fairness, accountability, and transparency*, 2020, pp. 648–657.

[23] 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.

[24] 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 *International Conference on Learning Representations*, 2019.

[25] C. Liu, *et al.*, "Jointly learning grounded task structures from language instruction and visual demonstration," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 1482–1492.

[26] 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," *2021 30th IEEE International Conference on Robot and Human Interactive Communication, RO-MAN 2021*, pp. 175–182, 2021.

[27] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

[28] K. E. Schaefer, *Measuring Trust in Human Robot Interactions: Development of the "Trust Perception Scale-HRI"*. Boston, MA: Springer US, 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 Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 2020, pp. 1–5.

[31] E. Coronado, F. Mastrogiovanni, B. Indurkhya, and G. Venture, "Visual programming environments for end-user development of intelligent and social robots, a systematic review," *Journal of Computer Languages*, vol. 58, p. 100970, 2020.