Robotics: Science and Systems 2022
New York City, NY, USA, June 27-July 1, 2022

Negative Result for Learning from Demonstration: Challenges for End-Users Teaching Robots with Task And Motion Planning Abstractions

Nakul Gopalan[§], Nina Moorman, Manisha Natarajan, Matthew Gombolay
Georgia Institute of Technology

Abstract—**Learning from demonstration (LfD) seeks to democratize robotics by enabling non-experts to intuitively program robots to perform novel skills through human task demonstration. Yet, LfD is challenging under a task and motion planning setting which requires hierarchical abstractions. Prior work has studied mechanisms for eliciting demonstrations that include hierarchical specifications of task and motion, via keyframes [1] or hierarchical task network specifications [2]. However, such prior works have not examined whether non-roboticist end-users are capable of providing such hierarchical demonstrations without explicit training from a roboticist showing how to teach each task [3]. To address the limitations and assumptions of prior work, we conduct two novel human-subjects experiments to answer (1) what are the necessary conditions to teach users through hierarchy and task abstractions and (2) what instructional information or feedback is required to support users to learn to program robots effectively to solve novel tasks. Our first experiment shows that fewer than half (35.71%) of our subjects provide demonstrations with sub-task abstractions when not primed. Our second experiment demonstrates that users fail to teach the robot correctly when not shown a video demonstration of an expert's teaching strategy for the exact task that the subject is training. Not even showing the video of an analogue task was sufficient. These experiments reveal the need for fundamentally different approaches in LfD which can allow end-users to teach generalizable long-horizon tasks to robots without the need to be coached by experts at every step.**

I. Introduction

Humans exhibit the ability to learn and solve long-horizon, multi-task problems. For example, a new warehouse employee can easily be taught to pack boxes with multiple types of objects and varying order specifications with a few demonstrations. Such a problem is multi-task, i.e., the number or types of objects in an order might be different, and *long-horizon*, as the worker solves a series of smaller sub-tasks such as picking the box, objects, packing materials; safely placing them; and packing the box, to complete an order. The field of Learning from Demonstration seeks to enable robots to exhibit a human ability to learn from end-user demonstration and scale the power of robotics. Yet, to this day, robots do not have a general-purpose ability to learn novel multi-task long-horizon tasks by demonstrations, despite prior work [4].

While prior work in LfD has shown that users can provide demonstrations at some level of abstractions, such as keyframes [1] or sub-task specifications for a hierarchical task network [2]. Study participants were given *explicit* instructions on how to teach the robot each task. Such an approach is untenable for scaling up to a vision of ubiquitous robotics, as it is impractical for experts to teach every end-user how to program robots each and every desired task. Instead, we examine how people teach novel tasks to robots in the absence of a roboticist's explicit tutelage. We first investigate whether people naturally prefer teaching using abstractions without explicit priming. Second, we want to test the efficacy of different modes of end-user instructions and feedback in soliciting correct sub-task based abstractions to solve robot learning on multi-task scenarios. We aim for these teaching modes to enable users to generalize their training towards teaching novel tasks without needing an expert to show them how. However, our novel experimental results demonstrate that roboticists are far from reaching this goal.

To enable humans to teach robots, the human-robot interaction (HRI) community has developed robot learning methods using Learning from Demonstrations (LfD) or programming by demonstrations as covered by multiple thorough surveys [4], [5]. LfD allows non-experts to teach tasks to robots. However, we do not want robots to solve the same repetitive tasks; we want robots to learn concepts that can be repurposed under novel specifications and under novel environmental settings. Accordingly the robot must learn concepts such as sub-tasks, skills or sub-goals, that can be repeated under novel task specifications. For example, a robot given a demonstration to pick up and stack three blocks on top of another should then be able to stack five blocks, with the object locations changed. The robot thus needs to learn a generalizable *abstraction* for stacking one block on top of another, which affords repeated application in the presence of varying block configurations from novel locations.

The HRI community has also proposed algorithmic solutions to learn sub-tasks from millions of demonstrations [6], [7], [8]. However, learning sub-task based decompositions algorithmically from few demonstrations is still an open problem as it is in the small data regime with high dimensionality. Learning good sub-task abstractions for tasks computationally is an open problem in reinforcement learning even with methods that require millions of learning episodes [9], [10]. To circumvent this theoretically and algorithmically challenging problem, we want to test the ability of humans to provide useful abstraction

[§]Corresponding Author: `nakul_gopalan@gatech.edu`

to robots. Furthermore, these abstractions should enable the robot to solve novel task and motion planning problems (TAMP) [11], affording generalization.

In this work we examine if humans *can teach* abstractions to agents and if these abstractions are *useful* to the agent for the purposes of planning in a Task and Motion Planning based formalism. We conducted two human-subjects experiments in which users are given the opportunity to teach robots via TAMP abstractions. In our first study ($n = 28$), we tested whether people can be primed to use abstractions and what factors induced people to use abstractions effectively. Our first experiment demonstrated that is challenging for users to provide sufficient abstractions for multi-task scenarios. This problem of providing sufficient abstractions relates to the question of the correspondence problem [12], where subjects do not know the characteristics of an optimal demonstration from a robot's perspective. Hence, we conducted a follow-up study ($n = 24$), where we tested different paradigms for soliciting demonstrations that provide sufficient abstractions to solve multi-task problems with long horizons. The paradigms that we tested included (1) a video of an analogue task to the one the subjects' solve, (2) a robot debugging demonstration that used the human's demonstration to attempt to solve a task when the object positions have changed, and (3) a video demonstration of an expert partially teaching the exact same task the subjects are attempting to teach the robot. The primary contributions of the paper are as follows:

1) We are the first to conduct human subjects experiments ($n = 28$ and $n = 24$) to compare strategies used by non-experts for teaching robots in a long-horizon, multi-task setting.
2) The results of our first study show that demographic factors, such as the IQ, of the participants affect the perceived workload and the ability to provide sub-task based abstractions ($p < 0.05$).
3) Results from our first study also show that the majority of the subjects (64.29%) do not naturally teach tasks with abstractions to robots. However, subjects improve at teaching tasks with abstraction when properly induced ($p < 0.05$).
4) Our results demonstrate that for each and every task only a video presenting an optimal teaching strategy allows 100% of the subjects to create necessary and sufficient abstractions, as opposed to only 25%, 25%, or 58.33% of participants when primed with text-based instructions, an expert demonstration on an analogue task, and a debugging demonstration on the robot, respectively. Our result highlights the difficulty in teaching a subject to provide abstractions *to teach novel tasks*, as it would be impossible to provide all end users with tutorials for the exact task they want to teach the robot.

II. Preliminaries

We use this section to provide definitions to ground the problem of interest.

Multi-modal tasks – A mode can be defined as a sub-manifold in which a robots motion is limited within because of the robot's contact with an object in the world [13], [14], [15], [16]. For example, when a robot is holding a cup, the range of motions it can perform is smaller than, and is within a sub-manifold of, the range of motions the robot can perform without the cup in its hand. Multi-modal tasks in robot planning are tasks where the modes in which the robot operates change during the execution of a task [11], [15], [16]. For example, our problem consists the robot picking and pouring materials used in gardening such as sand, lime, and manure. When picking sand the robot's scoop is empty and the robot is in a *transit mode* [11] until it reaches the sand. When pouring sand the robot is carrying sand and is in a *transfer mode*, where it transfers sand. During the transit mode the robot cannot lose contact with the sand until it is over the goal location.

Furthermore, *Long Horizon tasks* are tasks that have multiple mode switches during their execution. Task and Motion Planning [11] is a common model used to solve such long multi-modal problems. Within TAMP planning sub-tasks are divided based on the modes.

Task and Motion Planning – Task and Motion Planning (TAMP) problems are inherent to robotics, where a robot needs to perform discrete symbolic reasoning, and generate motion in the continuous state-action space without collisions. With Task planning, to solve the goal condition of stacking four blocks, the robot reasons about the location of each block, and checks to see how tall the stack is at any given time. Such planning does not consider the continuous nature of the world, and only considers the optimal sequence of blocks that need to be moved. Motion planning on the other hand attempts to move the robot without any collisions or breaking any existing constraints. For example consider the sub-task to place a block over an existing stack of blocks, the robot needs to move its arm over the existing block without colliding with the stack or the table, and without breaking any constraints, such as opening the gripper. TAMP problems are at the intersection of both of Task planning and Motion planning problems.

Sub-task-based abstraction – In this work we chose to learn a TAMP sub-task-based representation of the task. Each TAMP sub-task specification requires a pre-condition, a set of constraints, and a goal condition. The pre-condition is a predicate that tests if the sub-task can be used from a given state of the world. Constraints are specified to ensure collision free motion, and other requirements such as ensuring a closed/open gripper for grasping or maintaining the orientation of the arm when carrying objects. Finally, a sub-task needs a goal specification that accomplishes a sub-goal which helps complete the overall task specified. Given this sub-task specification a robot trajectory can be generated using constrained motion planning [17]. In this work, we investigate the conditions under which the end-users provide demonstrations that have valid pre-conditions, goal conditions, while maintaining consistent movement constraints during demonstrations to create valid TAMP sub-tasks.

We represent the constrained motion itself using Dynamic

Movement Primitives (DMPs) [18] learned from the data provided by the participants. DMPs are commonly used to represent learned trajectories because of their stability and sample efficiency [19], [20], [21], [22]. DMPs can be replaced with neural policies or a motion planner without any change to our formalism, but the issue of learning sub-tasks such that the constraints are demonstrated consistently across a sub-task demonstration remains the same across all these approaches. We are using DMPs for their sample efficiency as we are learning from real subjects.

A demonstrated abstraction with sub-tasks that have the consistent constraints throughout their execution are referred to as **Sufficient sub-task based abstractions**. In this work we investigate if people can teach sufficient sub-tasks to robots affording the robots generalizability to solve novel tasks.

III. Experiment design

We conduct two human-subjects experiments: (1) a 1×4 within-subjects experiment to test if users can be primed to provide sufficient sub-tasks and (2) a 1×4 mixed within-between-subjects experiment with different paradigms to teach users to provide sufficient sub-tasks. In this section, we first describe our research questions, our experiment domain, and the user interface, and we then provide additional details to set up the experiment.

A. Research Questions

We will first establish our research questions and then state our experimental design and study procedures. We formally state the following Research Questions (RQs):

- **RQ 1:** *Do people naturally provide abstractions for learning and planning?* Given that robots need people to provide sub-task based abstractions, we want to know whether people are already naturally primed to provide such demonstrations.
- **RQ 2:** *Can external factors/ inducements elicit abstraction-based teaching (e.g., ad nauseum repetition, or variation in task composition)?* We also sought to determine whether people naturally chose to use sub-task based abstractions to teach robots when faced with teaching tasks with numerous, repetitive components or a multi-task scenario where the robot has to solve different tasks in different instances for which the tasks share common sub-tasks.
- **RQ 3:** *Can participants be explicitly taught using textual descriptions of an analogue task to provide (more helpful) abstractions?* Requests to provide demonstrations using textual descriptions with figures for an analogue task are the simplest teaching guide. We wanted to see if these descriptions are enough to provide correct sub-task based demonstrations.
- **RQ 4:** *What demographic subject or objective factors and covariates influence how well people used abstractions for teaching?* We sought to examine if demographic covariates help people teach sub-task based abstractions to robots.
- **RQ 5:** *What explicit teaching guides if any might help the subjects learn to provide sufficient abstractions? Is this teaching guide generalizable to novel scenarios?* We created our second experiment specifically to answer this research question. We know that certain teaching guides work better than others when people are given direct instructions to teach robots with specific abstractions [3]. In this work, we seek to determine how little information about the current task needs to be provided to induce participants to provide correct sub-task based abstractions

B. Experimental Setup

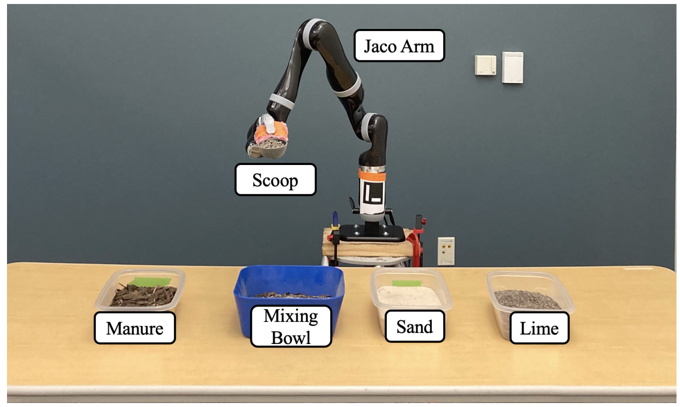

Fig. 1: Jaco robot setup: Subjects were required to teach the robot to create different types of soil mixtures in the mixing bowl using the sand, lime and manure available.

1) Task Domain: We designed a robot task domain setup in which we could create multi-task settings relatively easily with readily available raw materials. Hence, we consider a gardening task in which participants are required to teach the robot to create soil mixtures for different plants as shown in Figure 1. The setup consists of a robot arm, a pot of sand, a pot of manure, a pot of lime (calcium), and a mixing bowl. The subjects teach the agent to create soil mixtures required for three similar or different plants. The action space for this domain is continuous, and the locations of the objects are assumed to be known. The soil mixture domain allows for a multi-task setting and creates opportunities for constrained TAMP problems as described next.

2) Feasible Task and Motion Planning sub-tasks for the Soil Mixture Domain: In this section, we pick an example problem within our soil mixture domain: "create a soil mixture with one scoop of sand, and one scoop of manure." We then describe feasible abstractions that a user can provide to solve this task in our domain at different levels of granularity, going from coarser to finer grained TAMP abstractions. These abstractions span the breadth of sub-tasks that our users could demonstrate. We will also describe the relative merits of these abstractions for solving *all possible* tasks in the soil mixture domain.

Coarsest sub-task can be to create the complete soil mixture, one scoop of sand, and one scoop of manure, as a single sub-task, that is, use no abstractions at all when teaching as shown in Fig 2(a). The pre-condition for this sub-task would be that the scoop is empty. The constraints would be that the agent never collides and the goal condition would be to deposit one scoop of sand and one scoop of manure to the bin. However, given this sub-task, the robot can only solve tasks that are multiples of the base level task, e.g., four scoops of sand and four scoops of manure, but not all possible tasks.

A finer sub-task based abstraction would be to teach the robot to pick and pour one scoop of sand, and one scoop of manure as shown in Fig 2(b). The pre-condition for each task would be to have an empty scoop, the constraint would be to avoid collisions, and the goal condition would be to deposit the scoop of sand or manure into the bowl. This abstraction is a more generalizable sub-task abstraction as it allows the agent to solve novel tasks that are not present in the demonstrations given by the user. For example, once the robot learns how to pick and pour one scoop of sand and one scoop of manure, it can easily repeat these sub-tasks with the help of a *task planner* to solve the novel task of one scoop of sand, and four scoops of manure. The task planner is used to plan for the right sequence of sub-tasks that will complete the goal of a novel task. However, there is no constraint or condition in this demonstration that the robot picks up the scoop of sand or manure from the correct location, as the location of the sand or manure is not represented in the pre-condition, or the constraints, or the goal condition. If the location of the sand or manure were changed, the planner will ask the robot to perform the scoop gesture where the sand or manure was present during the demonstration and pour an empty scoop of sand or manure in the bowl, never satisfying the sub-tasks goal condition.

The finest feasible sub-task abstraction would be to teach the robot to pick sand and manure, and then teach the robot to pour the sand and manure as shown in Fig 2(c). To teach the pick sand sub-task, the pre-condition would be an empty scoop, the constraint would be to avoid collisions, and the goal would be to have sand in the scoop. To teach the pour task, the pre-condition would be to have a scoop with sand, the constraint would be to avoid collisions and the goal would be to drop the scoop of sand in the bowl. Such a sub-task demarcation allows the robot to understand the right pre- and post-conditions for each sub-task. Specifically, the robot learns to pick up the sand from any location as a sub-goal allowing the agent to pick up and pour objects to and from any location on the table. Moreover, the robot can again combine multiple pick and pour actions to deliver any required ratio of sand, and manure. This is a sufficient sub-task partition allowing the robot to solve the entire multi-task soil mixture domain with changing locations.

3) Robot Platform:

- Sawyer: For our first experiment, we used Sawyer, a seven degrees-of-freedom (DoF) arm from Rethink Robotics.
- JACO Arm: We switched to using a Kinova JACO seven (DoF) arm for the second experiment (Figure 1) because of mechanical failures on the Sawyer robot. Both robots can play back different demonstrated trajectories with high precision enabling non-expert users to teach the robot.

4) User Interface: We designed a user interface that allows participants to save demonstration trajectories and reuse them to solve tasks. The design of the interface was fine-tuned using iterative design methods during the pilot studies. The image of the interface is shared in the supplementary appendix A.2 [1]. The interface enables subjects to create and name sub-tasks, and then give a fixed number of demonstrations per sub-task. The sub-tasks can be reused by participants as many times as needed. Moreover, there is a procedure column for each occasion where subjects can create tasks by adding the demonstrated sub-tasks to the column sequentially to satisfy the occasion's task. The interface allows subjects to use abstractions, thereby creating shorter, repeatable, sub-tasks if so chosen.

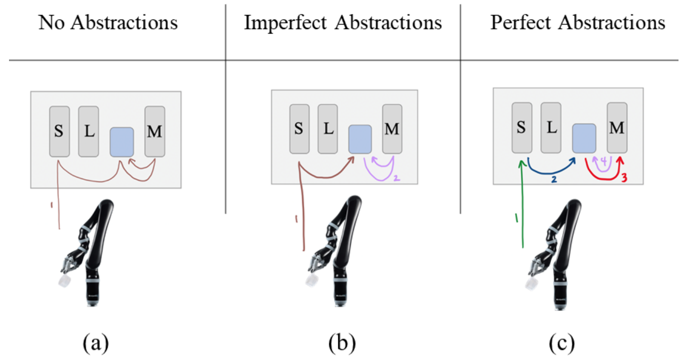

Fig. 2: An example of three different types of demonstration strategies to complete the task of "create a soil mixture with one scoop of sand, and one scoop of manure.". (a) If the subject gives a complete end-to-end demonstration as in the case of no abstraction, there is very little generalization to other novel tasks, e.g., creating a soil mixture with two scoops of sand and three scoops of manure. (b) If the participant breaks the task into sub-tasks where pick and place are a single sub-task unit, there is no constraint on picking materials, so if for example, the sand's location changes, the the agent cannot generalize solve tasks. (c) Breaking down the sub-tasks such that picking a material is a different sub-task and pouring a material is another sub-task. Such an TAMP abstraction is sufficient and can solve an undemonstrated novel task within the soil domain, without requiring the object locations being consistent.

All study participants were given equivalent training using the interface via a training video. Modulo latent confounders we have done our best to reduce confounds created by the interface itself during the experiment using iterative design, keeping the interface common across the conditions, and using training videos to provide equal training. Our results in Section V show that our interface and training were *sufficient* for participants to create abstractions. All our instruction videos and documents are provided in the supplementary website [1].

C. Experiment 1: Can subjects be primed or induced into providing sufficient abstractions?

In our first experiment, we investigate whether subjects are intrinsically motivated to provide abstractions when giving demonstrations to the robot or can be primed to do so.

[1] https://sites.google.com/view/rss-learning-sub-tasks-2022/home

We conducted a 1×4 within-subjects experiment with 28 participants (39.3% Female, Mean age = 21.42, Standard Deviation = 2.61) where the independent variable is the phase of the experiment. We also vary the order in which the phases are introduced to control learning effects. Each study phase corresponds to the type of task or amount of training the subject receives when teaching the agent. In each phase, the subject has to create three soil mixtures for different plants; we call these "occasions" in the study so the subject treats them as three distinct occasions of creating plant soil mixtures. The four phases are: the *Baseline* phase, the *Multi-task* phase, the *Large Number of Repeats* phase, and the *Multi-task via Written Instructions* phase.

- **Baseline (B):** Participants teach the agent a single task for three occasions with a few repetitions within this phase. The demonstration task for this phase involves creating a mixture of two cups of sand and one cup of manure.
- **Multi-task (MT):** The subject has to teach the agent different tasks for each of the three occasions in the MT phase. These tasks range from creating a soil mixture with the following number of scoops of objects for each of the three occasions: two of sand and one of lime, one of manure and one of lime, and one of sand and one of manure, respectively.
- **Large Number of Repeats (LR):** The subject has to teach the same task for each of the three occasions in the LR phase, but the task itself has a lot of repetitions within it. The task for this phase involved creating a soil mixture of ten scoops of sand and three scoops of lime.
- **Multi-task with Explicit Teaching via Written Instructions (MT+W):** The trainer gets explicit written instructions to use abstractions when training the agent. In the instructions, we describe abstractions in an unrelated task of cooking eggs. We also attempt to solicit correct abstractions by describing the robot's learning constraint in text.

As this is a within-subjects study with predictable learning effects across conditions, the ordering of the phases plays an important role in understanding subjects' ability to provide abstractions. Since, we want to study whether the participants naturally tend to provide abstractions or not, the subjects always begin with the baseline (B) phase. To establish which phase can induce abstractions faster we introduced the study phases to the participants in one of two possible orders. **Order 1**: B, MT, LR, MT+W. **Order 2**: B, LR, MT, MT+W. We only change the order with the MT and LR phases as giving instructions upfront, i.e., MT+W will bias the subjects to provide abstractions in the prior phases. We study the effect of introducing the two MT and LR in the results section V, RQ 2.

D. Experiment 2: Can teaching modes impact subjects ability to provide sufficient abstractions?

From our first study, we found that although participants learned to provide some form of demonstrations the participants generally failed to provide sufficient abstractions in those demonstrations to solve multi-task domains with constraints. Thus, we conduct a follow-up second experiment where we consider direct teaching modes: (1) a robot's debugging demonstration, (2) a video of an analogue task, and (3) an expert demonstration video of the same task the participants are teaching. The experiment is a 1×4 mixed within-between-subjects study with 24 subjects (45.83% Female, Mean age = 20.875, Standard deviation = 2.69). The experiment has three different teaching modes along with a baseline condition of no teaching. All participants experience the baseline phase and the phase which shows an expert demonstration video of the *same task* that the participants are teaching the robot. Moreover, half the participants observe the video demonstration of an analogue task, and the other half of the participants observe a debug demonstration that shows the consequence of their demonstration strategy. The experiment was constructed in this way to avoid learning effects between the teaching mode, and to avoid fatigue by keeping the length of the experiment to less than 2.5 hours. In all modes, the subject attempts to teach the robot in a multi-task scenario where each occasion has a different task. Moreover, the sand, lime, and manure pots' locations are changed between demonstrations. The object locations are changed to emphasize the need to teach constraint based sub-task with their demonstrations. The four conditions are as follows:

- **No teaching (NT):** Here no instructions are provided to the subjects. The subjects are free to use any strategy to teach.
- **Debug demonstration (DD):** The subjects are first provided with written instructions with diagrams showing sufficient abstractions in a similar task of touching two blocks. Further, the users are shown the consequence of the abstractions they provided in the "No Teaching" phase using a trajectory demonstration on the robot. Additionally, we also move the mixing bowl and the pot of sand to a new location. We used the demonstration strategies we observed in the first experiment to create these Wizard of Oz, debugging demonstrations. They have been designed to be informative about every sub-task a participant could have taught to successfully complete the overall task.
- **Video of analogue task (VA):** We provided the users with a video that demonstrated using our interface to teach the robot a related constraint based task of touching different blocks in a specific sequence.
- **Expert demonstration video of the Soil Mixture Task (EV):** We also wanted to see if providing a video demonstrating sufficient abstractions for parts of the soil mixing task would aid the subjects to extrapolate and provide sufficient abstractions for the entire task.

All participants started off with a phase of no teaching mode. They then either completed a debug demonstration or a video for an analogue task for their second phase in the experiment. Then all subjects finished with a final phase where they were shown a video showing parts of the soil mixture task. More details about the Experiment 2's conditions are provided in Appendix A.3. We know from prior work that showing a video tutorial for a task is sufficient to teach abstractions [3]. Hence, we chose to show videos of partial task solving towards the end to establish that people can provide sufficient abstractions with a little help from an expert in the problem domain without the complete solution. We conducted a between-subjects study, comparing a debugging demonstration and the video of an analogue task to prevent learning effects between the two modes, and to keep the study duration fatigue free for participants.

E. Study Procedure

Prior to the start of both the studies, we obtained approval for human-subjects experimentation from the Institutional Review Board at our affiliated institution. We recruited all participants through university mailing lists for both our studies. Due to the COVID-19 pandemic, we were unable to conduct large-scale user studies with off-campus participants. Nonetheless, we were able to recruit 28 and 24 participants for the first and second studies, respectively. All participants were compensated with a $25 and a $35 Amazon gift card for the first and second studies, respectively. No participant from the first study was allowed to take part in the second study. The procedure for both the user studies were quite similar and took a maximum of 2.5 hours to complete.

Upon arrival, the participants were asked to complete a pre-experiment questionnaire assessing demographic information, and pre-surveys that include the Big-5 personality test and their previous experience in teaching children or students. Subjects then participated in a practice round to get familiarized with the user interface while providing kinesthetic demonstrations to the robot. The experimenter then explained the soil-mixture task and the user interface to the participants. In the first study, the participants begin with the baseline condition, and follow either **Order 1** or **Order 2**. The order for each trial is chosen at random. After each phase, the participants also filled out a questionnaire for measuring their workload using the NASA-TLX [23]. At the end of teaching tasks for all the phases in the first experiment, the participants took an online IQ test [24].

In the second experiment, the participants follow the same pre-study procedures. Participants perform three soil mixture tasks with different teaching modes to help them provide sufficient abstractions. The subjects will begin with the NT condition, followed by either DD or VA (the between-subjects component). They will conclude with the EV condition. Participants also filled out the NASA-TLX workload questionnaire after each condition. The between-subjects variable (DD or VA) for each trial was randomized at the start of the trial.

IV. METRICS

We used the following metrics to measure the performance of the users teaching our robots.

- *Personality Questionnaire:* At the start of the study, participants filled out the the Big-5 personality questionnaire [25] on a five-point Likert scale.
- *Workload:* Participants filled the NASA-TLX questionnaire [23] to assess perceived workload for each condition.
- *IQ Metric:* We gave the participants in the first experiment an approximate open-source Intelligence Quotient (IQ) test [24] to test whether IQ has any relation to the ability to teach with abstractions. This test took each participant approximately 30 minutes to perform. We employ this IQ test as a proxy for measuring traits, such as reasoning skill, vocabulary and academic achievement [26], [27], [28]. We did not conduct the IQ test for the participants in the second experiment because the results from the first experiment answered the relevant research question.
- *Task Completion Time:* The duration of each phase was measured and was known to the participants.
- *Abstraction Score:* We also created an abstraction rubric to measure the performance of the participant in providing useful abstractions. The rubric provided a point for every valid TAMP abstraction provided by the user as described in Sec. III-B2. Moreover, a point is also provided for every valid TAMP abstraction that can be created from the abstractions provided by the user, i.e., if the user provided finer-grained TAMP abstractions, such as *picking sand* and *pouring sand*, the rubric also provided points for other coarser TAMP abstractions that can be satisfied by the finer abstractions, such as *pick and pour one cup of sand*. This scoring strategy is important, as if valid and generalizable low-level abstractions are provided to the robot it can solve more tasks. However, we do not to award points for extremely low-level abstractions, e.g. *move* $1\,\mathrm{cm}$ *to the left*, which would not be efficient in solving the task. All the valid abstractions that a user can provide to the robot in the soil mixture domain have been described in Sec. III-B2. We show in the Section V and with Fig. 4 that this is a valid scoring strategy to measure the generalizability of a given demonstration to solve a wide variety of tasks.
For example, in the task of making a soil mixture with two scoops of sand and one scoop of manure, if the participant gave a demo of the complete task, the demo would get 1 point. However, if the participant broke the task into creating abstractions of scooping one cup of sand and another for scooping one cup manure and use these constraint based sub-tasks to complete the overall task, then the demo would receive one point for a scoop of sand another for manure and one additional point to complete the overall task. Abstractions earn more points as breaking up the tasks in to sub-tasks help solve other tasks. The rubric does not award points for just taking a low-level action or teaching an unnecessary sub-task. To gain a point the created abstraction needs to create a sub-task based on a valid constraint.
- *Binary Abstraction Score*: We created a *binary score* where a participant's demo scored 1 if the demo had *any* abstraction in the phase and 0 if the demo did not.
- *Perfect Abstraction Score*: Finally, we checked whether the demonstrations given by the participants create valid sub-tasks with valid constraints. The participant's demonstrations were scored 1 if *sufficient* abstraction was provided in the phase, else 0. The participants were unaware of this rubric, and were told to complete the phases as efficiently as possible.

(a) Reaching for sand. (b) Picking up sand. (c) Move to bowl. (d) Dumping sand.

Fig. 3: Series of images demonstrating the plan created from learned trajectories on the robot to pick and drop sand from the set of 10 tasks created to test the learned policies from different demonstrations. For more details refer Appendix B.1.

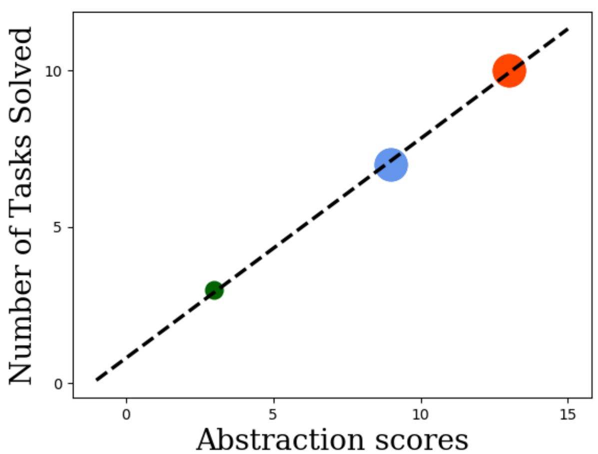

Fig. 4: Here we have plotted the abstraction score and the corresponding number of tasks *five* users' demonstrations were able to solve on the robot. There are a total of ten tasks and seven of these ten tasks are unseen to the robot previously. The larger circles indicate two users' demonstrations for the given score and tasks solved. The colors indicate the type of abstraction taught, green for no abstraction, blue for imperfect abstractions and red for sufficient TAMP abstractions. For a demonstration to be good it should be score higher, as the robot can solve all the tasks. As it can be seen from the plot these good demonstrations also have a large abstraction score. The dotted line is the linear regression of the scores vs the tasks solved, and it demonstrates that the abstraction scores that we created correlate well with the number of tasks that a given demonstration can solve.

V. Results

We first demonstrate that the abstraction score we created as described in the previous section is valid. We then present our investigations into the research questions posed in Sec. III-A.

A. Planning with the learned Policies on the Robot

We justify the creation of the abstraction score by comparing the task solving potential of trajectories demonstrated by our participants on the real robot. For this we created a set of 10 tasks to be solved by 5 demonstrations-sets chosen to represent different ranges of the abstraction score. This comparison shows that demonstrations that are given without sub-task abstractions can solve fewer than half the tasks. Specifically, tasks where the locations of the objects is changed arbitrarily can only be solved by the demonstrations in the highest quartile of the abstraction scores as observed in the multi-task phase with clear instructions in the first experiment. To measure this we create a set of ten tasks, in which seven tasks are completely novel, i.e., users did not provide any demonstrations for these seven tasks. Demonstration sets that do not use abstractions to train the robot are able to solve only the exact task that was taught to the agent, i.e., three out of ten tasks. With a slightly higher quality abstraction, where the users break apart tasks into picking and pouring individual scoops of sand, manure and lime, the robot can plan arbitrary combinations of these sub-goals, allowing the robot to solve seven out of ten tasks. The demonstration sets which were given keeping in mind that the robot learns using goals and constraints for sub-tasks, and separates the picking and pouring for scoops of objects solve all ten tasks even when object locations change. This experiment primarily demonstrates that our abstraction scoring system was practical, and higher abstraction scores for demonstrations indicate the ability to solve a larger number of possible tasks. Figure 4 shows that demonstrations that solve more novel tasks on the robot, also have high abstraction scores. Hence, our abstraction scores are a valid measure a demonstration's quality in solving novel tasks using the TAMP formalism. The complete experimental details are provided in the Appendix B with the whole set of tasks and their outcomes in Table 1 of the appendix. An example of the trajectory is shown in Figure 3 along with a video supplement showing multiple trajectories[2]. These empirical results validate that our abstractions score quantifies the capability of a given demonstration to generalize to novel tasks. Next we will discuss our Research Questions and their implications.

B. Research Questions

RQ1: *Do people naturally provide sufficient abstractions for learning and planning?* Results from our first study using the Binary Score indicate that only 35.71% of the participants used

[2]https://sites.google.com/view/rss-learning-sub-tasks-2022/home

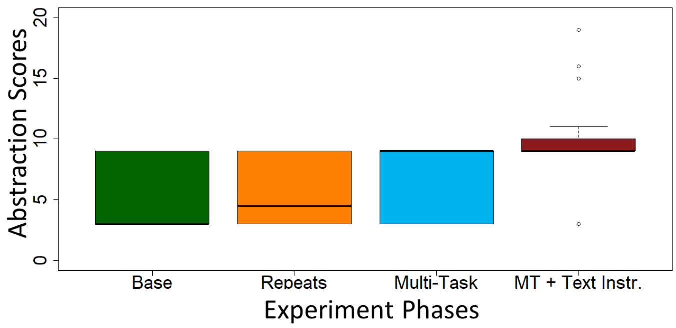

Fig. 5: Box plot indicating the abstraction score distributions for the phases of Baseline, Large number of repeats, multi-task, multi-task with instructions, respectively for the first study. As soon as participants' are directly asked using textual instructions to teach using sub-task based abstractions, in the multi-task with instruction phase, majority of the participants choose to do so, but they still fail to provide optimal sub-task based abstractions.

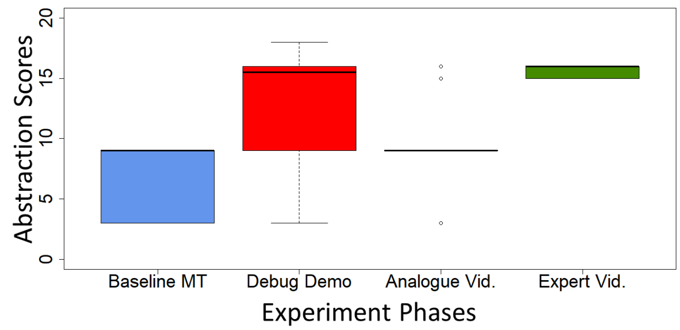

Fig. 6: Box plot indicating the abstraction score distributions for the phases of Baseline Multi-task, Debug Demo, Analogue Video, and Expert training demonstration, respectively for the second study. Note that the expert training demonstration video that shows a partial solution performs much better than other modalities to train subjects.

any abstraction in the baseline phase, implying that the majority of the participants do not provide abstractions naturally.

Takeaway: We posit that the majority of subjects have difficulty in knowing where to break a task in a continuous robot domain, as there is no natural indication of what a sub-task for a robot could be.

RQ 2: *Can external factors or inducements elicit abstraction-based teaching?* In our first study, we examine the effectiveness of using different priming methods to help subjects use abstractions while providing robot demonstrations. A Wilcoxon-signed rank test with abstraction score as the dependent variable and study phase as the independent variable shows that there exists no statistical difference in abstraction scores between the **LR** phase and the **MT** phase. Further, we conducted a Cox-Regression Hazard analysis to verify if task order might be critical in determining the number of abstractions a user provides, but did not find any significance.

Takeaway: Our results imply that seeing a large number of repetitions and a multi-task setup both encourages people to use abstractions at similar rates.

RQ 3: *Can participants be explicitly taught to provide (more helpful) abstractions?* In the robot study 24 out of the 28 (85.71%), participants learned to teach abstractions to the agent after **MT+W** phase (measured with binary abstraction score). The remaining four participants could not learn to break tasks apart to teach the robot. The most common form of abstraction chosen was to "pick and pour sand," "pick and pour manure," and "pick and pour lime." When tested against the **MT+W** phase where explicit instructions were given to break down tasks into repeatable sub-goal based abstractions, participants succeeded in providing abstractions and performed significantly better. We ran multiple Wilcoxon-signed rank tests with Bonferroni correction ($\alpha = 0.05/6$) to compute pairwise comparisons for abstraction scores across the different study phases in the first study. Results from the Wilcoxon-signed rank tests indicate that abstraction scores from the **MT+W** were significantly better than the baseline ($Z = 165$, $p < 0.0001$), **LR** ($Z = 598$, $p < 0.001$), and **MT** ($Z = 560$, $p < 0.001$) with effect sizes $0.779, 0.740, 0.697$; respectively. The box-plot of all the abstraction scores are in Figure 6.

Although 24 participants were able to provide abstractions in the MT+W phase, only 7 out of the 28 participants provided demonstrations for sufficient sub-tasks according to the perfect abstraction score, despite being given clear instructions that the robot cannot touch two objects in the same trajectory as these are different goal constraints. **Takeaway:** From the first experiment, we note that a majority $(24/28)$ of the participants were able to provide task abstractions after being primed with the MT+W phase. However, only a small fraction $(7/24)$ of the participants were able to provide sufficient abstractions. These results show that teaching long horizon and multi-task problems to robots is not trivial, and correspondence problems between the robot and its teacher can be an issue in robot teaching.

RQ 4: *What demographic subjective or objective factors and covariates influence how well people used abstractions for teaching?* To analyze which subjective and objective factors play a significant role in influencing a user's ability to provide abstractions, we created a linear mixed effects model with abstraction score as the dependent variable, and the independent variables being study phases, conditions, with covariates of age, IQ, and personality score. We pick the model with the lowest Akaike information criterion (AIC) by pruning variables, and covariates from the largest possible model. All of the models were tested for normality and homoscedasticity for which the details are in the supplementary Appendix A.1[3].

For the first experiment, we found the abstraction score was significantly dependent on the phase of the study ($F(3, 112) = 25.05, p < 0.001$) and the IQ of the participants with ($F(1, 112) = 6.81, p = 0.01$).

[3]See footnote 1

Condition 1	Condition 2	p value	Effect of size
EV	VA	< 0.005	0.863
EV	NT	< 0.005	0.829
VA	NT	< 0.01	0.757
DD	NT	< 0.005	0.829

TABLE I: p-values for pairwise comparison of teaching modes on abstraction scores of subjects.

Takeaway: Our analyses indicate that the user's ability to provide task abstractions is significantly dependent on the study phase and IQ.

RQ 5: *What explicit teaching strategies, if any, might help the subjects learn to provide sufficient abstractions? Is this teaching generalizeable to novel scenarios?* We computed six Wilcoxon-signed rank with Bonferrroni Correction ($\alpha = 0.05/6$) tests for pairwise comparisons of perfect abstraction scores across all combinations of the teaching modes used in the second experiment. The significant results from our pairwise comparisons are listed in Table I.

We also compare the ability for participants to provide the right sub-task decomposition (or abstractions) after the final phases of the first experiment (**MT+W**) and the second experiment **EV**, with a Wilcoxon-signed rank test and find that the abstraction scores of participants in the **EV** condition are significantly better ($Z = 568$, $p < 0.001$, effect size=0.616).

Takeaway: Our results indicate that **EV** is the most effective technique in eliciting sufficient abstractions from non-experts for teaching a robot in a multi-task, long horizon setting. However, this approach does not scale well to novel tasks that an end-user might want to teach a robot. These results imply that showing the participants a video of the expert demonstrating the training to teach the same task that the participant is teaching is better than other teaching modalities to help the robot learn to solve novel tasks. However, providing such videos for a household-hold robot would not be possible for all cases.

Note on Perceived Workload and Abstractions: To analyze how providing abstractions can affect the perceived workload of a user, we ran a Wilcoxon-Signed rank test with workload as the dependent variable. Our results show that workload was significantly dependent on the interaction effect between abstraction scores ($F(3, 112) = 11.48, p < 0.001$), and the phase of the study, with a linear effect from the IQ of the participant ($F(1, 112) = 5.29, p = 0.02$) for the first experiment. However, the perceived workload was not dependent on any of the independent variables with significance in the second experiment. We hypothesize that this is because all the phases had multi-task scenarios, reducing the significance of variables such as the study phase or ordering, to predict workloads.

VI. Related Work

Learning from demonstration (LfD) is a ubiquitous approach for enabling humans to program robots to perform new skills via human task demonstrations [29], [30], [31], [4]. Prior work in LfD has learned impressive dynamic skills on the robot [29], [30], and the ability to play high-dimensional games [32]. These approaches generally attempt to either directly model the robot's unknown policy [31] or infer the robot's latent reward function [33], [34], [35]. Some LfD approaches attempt to acknowledge the way humans teach tasks by modelling feedback more accurately [36], [37]. However, these works have not addressed the question of whether people teaching agents tasks using an abstraction hierarchies.

In the HRI community significant research has shown that people can teach abstractions, sub-tasks or otherwise, when tutored to teach the exact same task [3], [2]. Cakmak et al. [3] attempt to teach keyframe based abstractions when subjects are show a video tutorial of the same task. Mohseni et al. [2] attempt to teach hierarchical task networks from human feedback with a well designed interface and training to use the interface. Multiple works have shown that novice users can learn to use their interface and teaching paradigms effectively to train the agent [8], [6], [7]. These methods generally have an algorithmic contribution as well where the algorithm can learn to separate tasks or learn constraints from data [6], [7], [22].

However, these methods usually extract tasks from low-dimensional torque and environment data with sophisticated statistical techniques that are computationally expensive and might not generalize to novel environments and tasks. Instead, we seek to empower end-users to train these robots using sub-tasks. People can remove this computational bottleneck and provide correct sub-tasks as they have better generalization capabilities than modern robotics techniques. Hence, we test whether the users are equipped to provide such demonstrations, and what type of priming or tutoring would elicit demonstrations using sub-tasks that help the robot to generalize to novel task specifications while keeping in mind that the robot is going to solve a TAMP [11] problem, and that the sub-tasks specified should be usable by a TAMP formalism. We note that our focus is not on user interface design unlike previous works [3], [2], [6], [8]. Rather, we are keen on investigating priming mechanisms and teaching guides to help users teach useful sub-task based abstractions given a sufficient interface.

Hierarchical learning and abstraction have been of active areas of research [38]. These hierarchical abstractions have been shown to be more efficient than learning [39] or planning [11] at the ground level in the continuous or low-level state space of the agent. In hierarchical planning and/or learning formalisms, the abstraction hierarchy is pre-specified and the policies for the abstractions and over abstractions are learned by the agent [40], [39], [41], [42]. These methods have been used to solve challenging robotics problems, such as playing table-tennis where each stroke is a different skill [43], and to discrete domains, such as Taxi [39]. Approaches that attempt to learn both the hierarchy and the policies for the agents generally work in simulation [9], [10], but can require significant expert knowledge about the problem domain in robotics [44], preventing wide adoption of these ideas. Incorrect specification of a hierarchy for learning has been shown to hurt learning instead of helping [45].

VII. Discussion

Our novel human-subject experiments show that people can be trained to teach tasks using sub-tasks to a robot. However, it is hard to train *everyone* to teach robots sub-tasks for *any* task. Roboticists cannot be expected to create instructional videos for every possible task that a person might attempt to solve.

A key issue here is the inability of majority of the participants to teach correct abstractions to the robot when asked to with a debugging demonstration or a video of a related task. People can break apart tasks in general, as shown in our first experiment, but these might not be robot usable abstractions. Such issues have been raised before with the correspondence problem [12] where people's body parts do not match a robot's parts. Similarly, a robot's learning methodology is not akin to an human's learning methodology. This discrepancy might lead people to teach agents tasks incorrectly.

These experiments also show that teaching novel long horizon tasks to robots is non-trivial and robot learning has to still make large progress in the realms of sample complexity of learning, and understanding how humans can teach agents. We also hypothesize that a curriculum where participants observe their teaching outcomes over a series of tasks might help participants help teach other novel tasks to the robot.

In summary, we provide these key takeaways as design guidelines for researchers in machine learning, planning, and human-centered design:

1) Non-expert humans do not automatically teach through abstractions – let alone abstractions sufficient for an intelligent agent to leverage in a TAMP setting. As such, researchers must be careful not to make this assumption.
2) Explicitly teaching humans to use sub-tasks to train robots does not work sufficiently well if the teaching is done using textual descriptions. Nor are videos of expert demonstrations of a related task or showing a robot's failed debugging demonstrations.
3) Providing an expert demonstration video for the *same task* that the participants are teaching allows them to provide sufficient abstractions 100% of the time, but such an approach does not scale to support novel tasks.
4) Demographic factors may impact the ability of a person to provide helpful abstractions. Further research is needed to characterize this phenomena to ensure equitable access to the benefits of learning agents.

VIII. Limitations

Finally, we seek to address key limitations in our work. First, our sample population consisted primarily of college students, which may not represent the broader population. We also note that our measure of IQ is imperfect, as it was performed with an open-source, online test rather than a trained, in-person examiner. Finally, we believe there are important mediating effects between measured IQ, workload, and score, where are difficult to isolate due to assumptions of available statistical procedures for mediation analysis.

IX. Conclusion

In this paper, we investigated whether non-expert humans are capable of specifying high-quality sub-tasks when teaching robot agents to perform multi-step tasks. The assumption of readily available, informative sub-task abstractions is ubiquitous in the machine learning and planning communities. We conducted two novel human subject experiments. In our first experiment, we show that majority of people do not naturally provide high-quality abstractions when teaching a learning agent. Our second experiment shows that people can be trained to provide sufficient sub-tasks for planning a 100% of the times, if they are shown a part of an expert teaching demonstration of the task that they are attempting to teach the robot. While other more generalizable approaches are not as successful in soliciting sufficient sub-tasks. Providing such expert demonstrations is not feasible for all tasks. Our results provide important guidance to the research community (1) not to rely on non-expert humans to readily provide sufficient abstractions and (2) renew research into human-centered design around robot learning to democratize robot teaching for non-experts.

X. Acknowledgments

This work was sponsored by MIT Lincoln Laboratory (7000437192), NASA Early Career Fellowship (80HQTR19NOA01-19ECF-B1), the National Science Foundation (20-604), and a gift from Konica Minolta, Inc. The content of this work is solely the responsibility of the authors and does not necessarily represent the official views of our sponsors.

References

[1] Baris Akgun, Maya Cakmak, Karl Jiang, and Andrea L Thomaz. Keyframe-based learning from demonstration. *International Journal of Social Robotics*, 4(4):343–355, 2012.

[2] Anahita Mohseni-Kabir, Charles Rich, Sonia Chernova, Candace L Sidner, and Daniel Miller. Interactive hierarchical task learning from a single demonstration. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, pages 205–212, 2015.

[3] Maya Cakmak and Leila Takayama. Teaching people how to teach robots: The effect of instructional materials and dialog design. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 431–438, 2014.

[4] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[5] Aude Billard, Sylvain Calinon, Ruediger Dillmann, and Stefan Schaal. Survey: Robot programming by demonstration. Technical report, Springrer, 2008.

[6] Eric M. Orendt, Myriel Fichtner, and Dominik Henrich. Robot programming by non-experts: Intuitiveness and robustness of one-shot robot programming. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 192–199, 2016.

[7] Yoan Mollard, Thibaut Munzer, Andrea Baisero, Marc Toussaint, and Manuel Lopes. Robot programming from demonstration, feedback and transfer. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 1825–1831. IEEE, 2015.

[8] Chris Paxton, Andrew Hundt, Felix Jonathan, Kelleher Guerin, and Gregory D Hager. Costar: Instructing collaborative robots with behavior trees and vision. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 564–571. IEEE, 2017.

[9] Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[10] Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. In *International Conference on Machine Learning*, pages 3540–3549. PMLR, 2017.

[11] Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Integrated task and motion planning. *Annual review of control, robotics, and autonomous systems*, 4:265–293, 2021.

[12] Chrystopher L Nehaniv, Kerstin Dautenhahn, et al. The correspondence problem. *Imitation in animals and artifacts*, 41, 2002.

[13] Rachid Alami, Thierry Siméon, and Jean-Paul Laumond. A geometrical approach to planning manipulation tasks. the case of discrete placements and grasps. In *International Symposium on Robotics Research*, 1991.

[14] Rachid Alami, Jean-Paul Laumond, and Thierry Siméon. Two manipulation planning algorithms. In *Proceedings of the Workshop on Algorithmic Foundations of Robotics*, 1995.

[15] Kris Hauser and Jean-Claude Latombe. Multi-modal motion planning in non-expansive spaces. *The International Journal of Robotics Research*, 29(7), 2010.

[16] Kris Hauser and Victor Ng-Thow-Hing. Randomized multi-modal motion planning for a humanoid robot manipulation task. *The International Journal of Robotics Research*, 30(6), 2011.

[17] Mike Stilman. Task constrained motion planning in robot joint space. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3074–3081. IEEE, 2007.

[18] Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):328–373, 2013.

[19] Jan Peters and Stefan Schaal. Natural actor-critic. *Neurocomputing*, 71(7-9):1180–1190, 2008.

[20] Zhijun Li, Ting Zhao, Fei Chen, Yingbai Hu, Chun-Yi Su, and Toshio Fukuda. Reinforcement learning of manipulation and grasping using dynamical movement primitives for a humanoidlike mobile manipulator. *IEEE/ASME Transactions on Mechatronics*, 23(1), 2017.

[21] Scott Niekum, Sarah Osentoski, George Konidaris, and Andrew G Barto. Learning and generalization of complex tasks from unstructured demonstrations. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012.

[22] Nakul Gopalan, Eric Rosen, George Konidaris, and Stefanie Tellex. Simultaneously learning transferable symbols and language groundings from perceptual data for instruction following. In *Proceedings of Robotics: Science and Systems XVI.*, 2020.

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

[24] FSIQ. Open source psychometrics project Full Scale IQ Test. https://openpsychometrics.org/tests/FSIQ/. Accessed: 2021-05-28.

[25] Oliver P John, Sanjay Srivastava, et al. *The Big-Five trait taxonomy: History, measurement, and theoretical perspectives*, volume 2. University of California Berkeley, 1999.

[26] Wayne Weiten. *Psychology: Themes and variations*. Cengage Learning, 2021.

[34] Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *ArXiv*, abs/1710.11248, 2018.

[27] Ian J. Deary, Steve Strand, Pauline Smith, and Cres Fernandes. Intelligence and educational achievement. *Intelligence*, 35(1):13–21, 2007.

[28] Adam Hampshire, Roger R Highfield, Beth L Parkin, and Adrian M Owen. Fractionating human intelligence. *Neuron*, 76(6):1225–1237, 2012.

[29] Kai Ploeger, M. Lutter, and Jan Peters. High acceleration reinforcement learning for real-world juggling with binary rewards. *ArXiv*, abs/2010.13483, 2020.

[30] Letian Chen, Rohan Paleja, and Matthew Gombolay. Learning from suboptimal demonstration via self-supervised reward regression. In *Proceedings of the Conference on Robot Learning*, 2020.

[31] Jonathan Ho and S. Ermon. Generative adversarial imitation learning. In *NIPS*, 2016.

[32] Mikayel Samvelyan, Tabish Rashid, C. S. D. Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, P. Torr, Jakob N. Foerster, and S. Whiteson. The starcraft multi-agent challenge. In *AAMAS*, 2019.

[33] P. Abbeel and A. Ng. Apprenticeship learning via inverse reinforcement learning. *Proceedings of the twenty-first international conference on Machine learning*, 2004.

[35] Brian D. Ziebart, Andrew L. Maas, J. Bagnell, and A. Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, 2008.

[36] J. MacGlashan, Mark K. Ho, R. Loftin, Bei Peng, G. Wang, D. L. Roberts, Matthew E. Taylor, and M. Littman. Interactive learning from policy-dependent human feedback. In *ICML*, 2017.

[37] W. B. Knox and P. Stone. Interactively shaping agents via human reinforcement: the tamer framework. In *K-CAP '09*, 2009.

[38] George Konidaris. On the necessity of abstraction. *Current opinion in behavioral sciences*, 29:1–7, 2019.

[39] Thomas G Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of artificial intelligence research*, 13:227–303, 2000.

[40] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[41] Nakul Gopalan, Michael Littman, James MacGlashan, Shawn Squire, Stefanie Tellex, John Winder, Lawson Wong, et al. Planning with abstract markov decision processes. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 27, 2017.

[42] K. Erol, J. Hendler, and D. Nau. Htn planning: Complexity and expressivity. In *AAAI*, 1994.

[43] Katharina Muelling, Jens Kober, and Jan Peters. Learning table tennis with a mixture of motor primitives. In *2010 10th IEEE-RAS International Conference on Humanoid Robots*, pages 411–416. IEEE, 2010.

[44] George Konidaris and Andrew Barto. Skill discovery in continuous reinforcement learning domains using skill chaining. *Advances in neural information processing systems*, 22:1015–1023, 2009.

[45] Nicholas K. Jong, Todd Hester, and Peter Stone. The utility of temporal abstraction in reinforcement learning. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '08, page 299–306. International Foundation for Autonomous Agents and Multiagent Systems, 2008.