Discovering Task Constraints Through Observation and Active Learning

Bradley Hayes[1] and Brian Scassellati[1]

Abstract— **Effective robot collaborators that work with humans require an understanding of the underlying constraint network of any joint task to be performed. Discovering this network allows an agent to more effectively plan around co-worker actions or unexpected changes in its environment. To maximize the practicality of collaborative robots in real-world scenarios, humans should not be assumed to have an abundance of either time, patience, or prior insight into the underlying structure of a task when relied upon to provide the training required to impart proficiency and understanding. This work introduces and experimentally validates two demonstration-based active learning strategies that a robot can utilize to accelerate context-free task comprehension. These strategies are derived from the action-space graph, a dual representation of a Semi-Markov Decision Process graph that acts as a constraint network and informs query generation. We present a pilot study showcasing the effectiveness of these active learning algorithms across three representative classes of task structure. Our results show an increased effectiveness of active learning when utilizing feature-based query strategies, especially in multi-instructor scenarios, achieving better task comprehension from a relatively small quantity of training demonstrations. We further validate our results by creating virtual instructors from a model of our pilot study participants, and applying it to a set of 12 more complex, real world food preparation tasks with similar results.**

I. INTRODUCTION

For the vast majority of situations, it is unreasonable to assume that a robot can be deployed in an environment carrying the full knowledge required to adequately perform its duties. As a motivating example, consider a household robot that is tasked with assisting a human with cooking duties. The preparation and treatment of each ingredient type likely requires its own trained skill (e.g., mashing potatoes vs. kneading dough). It would be impossible to precompute all possible skills required for a robot to have proficiency across all food preparation tasks, and equally difficult to develop objective functions enabling a robot to practically learn these skills on its own. Requiring a robotics expert to provide new programming for each shift in responsibilities, process, or task is often prohibitively expensive from both a temporal and monetary perspective. Further, it cannot be reasonably expected that the programmer will be a subject-matter expert for the target domain, resulting in the difficult situation of attempting to translate an expert's knowledge and learned heuristics into code.

To overcome this difficulty, Learning from Demonstration (LfD) has been developed and validated as both a popular and effective mechanism to afford non-experts the ability to easily impart new task and skill knowledge to robots [1], [3]. The ultimate goal of LfD research is to build systems capable of learning and generalizing tasks from naturally demonstrated examples by non-technical users. While many methods of skill learning can perform quite well and are considered feasible even when requiring thousands of simulated trials to converge on a proficient skill policy [24], it is clear that the applicability of LfD will be severely limited by such large training set requirements. It would be an unwise design decision to assume a human is willing or able to demonstrate dozens of executions of each possible cooking task for the sake of training a robot.

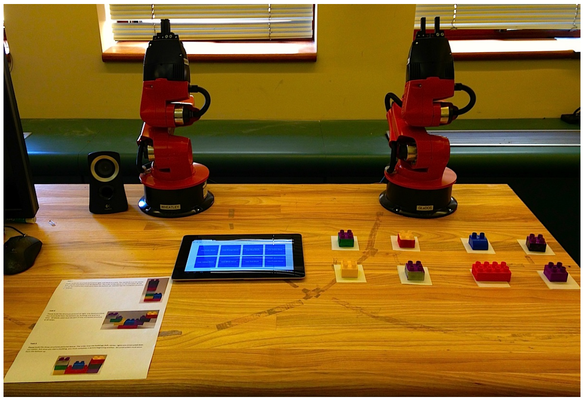

Fig. 1. Collaborative Workbench platform with experimental setup, used to learn task structure from demonstrations of construction activities.

Building more capable, intuitively trainable robots is a vital step towards transitioning them from isolated, independent workers to capable, safe, efficient collaborators. To facilitate this transition, the robotics community must collectively overcome a broad variety of challenges spanning the domains of skill acquisition, implicit and explicit communication [4], [22], solo and joint task understanding, and collaborative execution [11], [12].

When seeking to integrate collaborative robots into complex roles with non-trivial responsibilities, developing team-oriented behaviors becomes increasingly important. To this end, researchers have developed planning algorithms that, given the constraints of a task, allow for rapid planning and ideal resource allocation [21], [23]. As robots become more integrated into human environments, it will be equally important to develop mechanisms by which a robot can leverage task structure comphrension to synchronize its execution preferences with the expectations of its teammates [14], [18] and incorporate LfD-based skills into high-level planning systems such as Hierarchical Task Networks [17], [19].

[1]Computer Science Department, Yale University, 51 Prospect Street, Connecticut, USA `{bradley.h.hayes, brian.scassellati}@yale.edu`

Enabling these behaviors is contingent upon the agent having an understanding of the effects its actions have on the world. Skills acquired via LfD do not always have this knowledge accessible via the necessary symbolic formalization, as generating it requires considerable sensing or intention recognition abilities [16]. These effects can be dealt with indirectly, as skills may be viewed merely as functions acting on the environment, producing a new environment state given an existing one. Thus, learning which compositions of these functions are valid allows the agent to plan and act without direct symbolic representations of each component skill [15]. The problem of discovering these valid sets of compositions is equivalent to learning the constraint network for a given task.

In this work, we address the problem of a robot (Figure 1) learning the constraint network of a task through the observation of human instructors. Learning task structure in this manner can require a large set of demonstrations to achieve a sufficient amount of training diversity. If an instructor behaves habitually and does not produce novel demonstrations, very little new structural information can be acquired per training session and no new constraints will be learned/removed. As an example, if an instructor were teaching a robot about salad preparation but only ever chopped carrots before chopping celery, the robot will learn an artificial constraint indicating that carrot preparation must occur prior to celery preparation in this task. Further complicating matters, instructors are expected to properly modulate the diversity of each demonstration to the learner, maintaining responsibility for maximizing learning gains at each step. To mitigate these issues, a robot can leverage its embodiment to participate in the learning process, asking the instructor questions that seek to maximize its learning gains. This process, called Active Learning, has been shown to be effective in LfD domains [7].

Our work offers the following contributions:

- Two graph feature-based active learning query generation algorithms that significantly outperform a standard exploration function in constraint learning.
- A pilot study indicating how different active learning query strategies affect the learning of three representative classes of task constraint networks.
- An examination of the effects of drawing demonstrations from multiple instructors on learning task structure.
- An experiment showing that better training data may be obtained by optimizing for inspiring diverse instruction from a small group of instructors rather than extensive training from a single instructor.

II. Approach

To accomplish this, we use a graph transformation function that is applicable to the commonly used [2], [10], [9], [13] Semi-Markov Decision Process (SMDP) [20] task representation. The resulting SMDP dual graph (figure 2), hereafter referred to as the *action-space graph*, affords advantages to a robot learner in terms of expediting task structure comprehension. We show that the action-space graph can be used within the context of active learning to achieve accelerated understanding of a complex task structure given a realistic and limited set of demonstrations and instructors. We focus particularly on learning a constraint network for each task, indicating acceptable skill sequences to reach goal states for the activity. In doing so we remove the need for complex intention recognition or symbolic formalization of skills, making this approach particularly accessible to LfD-based systems with minimal high-level sensing requirements.

We focus especially on the domain of tasks and environments where instructor availability is limited (low demonstration count), human-robot communication is heavily constrained (no explicit communication from human to robot), and collaboration with other trusted robot instructors is limited or unavailable (low collaborator count). We maintain these very conservative restrictions in an effort to remain relevant to a maximally diverse range of collaborative robots.

III. Learning Environment

A. Task Learning Domain

This work focuses on using active learning and LfD to discern the underlying structure of a task without context. We represent a task SMDP using standard graphical notation $G = \{V, E\}$. We define an observed task execution x as a simple directed graph with membership in the set X consisting of all valid task execution graphs. Thus, $x = \{V_x, E_x\} \in X$ describes a single path through a set of vertices in V representing environment states linked by edges labeled with known, executed actions, terminating in a vertex describing a goal state (terminal vertex). We define a task as the weakly connected directed multigraph $T = \{V, \bigcup_{i=1}^{|X|} edges(x_i) \in X\}$, describing a Semi-Markov Decision Process. A vertex in this graph is labeled with information representing the state of the environment, while directed edges connecting environment states are labeled with actions that must be performed to transition between them. This typical SMDP representation is subsequently referred to as an *Environment-space Graph* (Figure 2—top).

Formally, the task structure discovery problem can be formulated as one of graph exploration. The learner is presented with an ever-growing library of observations L of successful skill execution sequences ($l \in L \implies l \in X$), successively gaining more insight into the task being performed as novel elements are added. Certain paths through the graph are more valuable than others, as a demonstration primarily serves to reveal new vertices and edges. As such, a learner is motivated to encourage an instructor to provide the most informative path per demonstration, desiring the instructor to show a demonstration x according to

$$\arg\max_{x \in X} |unique\_edges(L \cup x)|$$

B. Multi-Instructor Task Learning Domain

As robots become increasingly ubiquitous, the availability and magnitude of potential gains from distributed task learning across multiple instructors increases.

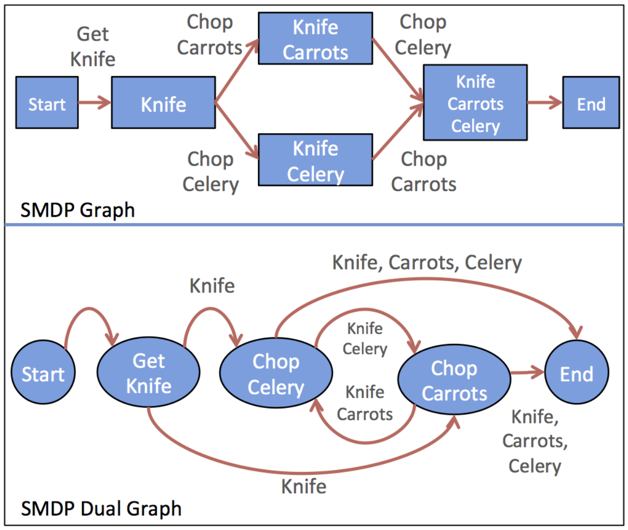

Fig. 2. Visualization of graph transforms on an everyday task. For this task, the knife must be acquired prior to chopping, though the order that the carrots or celery is chopped does not matter.

It is likely there will be a substantial overlap of training needs across collaborative robots. Across every robot trained in an instance of the general cooking task, some subsets of them will be trained for the same particular instantiations of the activity. Leveraging the data from these instructors with similar robots, environments, and task demands can be mutually beneficial, but also potentially dangerous if the shared data is invalid or maliciously constructed. For this reason, even as robots become commonplace in many task domains and the amount of relevant collaborators increases, it may not be a safe assumption that all others' training data should be implicitly trusted. This reinforces the need to expect situations in which there may be limited numbers of available collaborators. We define the multi-instructor task learning domain as m independent instances of the *task learning domain* whose observation libraries $L_1, L_2, ..., L_m$ can be joined and re-distributed to each participating instructor, leveraging not only additional demonstrations, but benefitting from the diversity inherent to multiple teaching styles. In our work, this step occurs subsequent to all training activity to avoid the assumption that collaborator data was available during the initial training of a task.

C. Activity Description

Our work focuses on learning the underlying structure of a collection of representative construction tasks. These tasks are defined through ordering and prerequisite conditions on a set of eight uniquely colored interlocking blocks (Figure 1). Block construction tasks were chosen as they can be flexibly or rigidly defined and will not bias participants towards particular demonstrations in a way that tasks involving familiar objects might. Participants were only permitted to construct from the bottom up in their demonstrations, resulting in a total possible task space of 40320 block usage sequences. The three representative tasks chosen to test our active learner query algorithms are drawn from three distinct structural compositions (Figure 3).

D. Workspace

The workspace for performing tasks on the Collaborative Workbench contained one KUKA YouBot arm, eight uniquely colored building blocks, a piece of paper with task descriptions, and a tablet computer (Figure 1). The leftmost YouBot arm was not utilized for this experiment. Each of the eight blocks had a labeled place-card designating its home position affixed to the table within range of the arm (the "resource area"). The tablet, placed next to the resource area, presented an interface with buttons corresponding to each of the blocks on the table. The area immediately in front of the participant was informally designated as the construction area. The paper with task descriptions and reference images was placed just left of the construction area.

IV. Learning System

A typical passive learner would gain knowledge about the structure of a task by observing performances of it. Duplicate observations would not provide new insight into the underlying structure, though they may be helpful for determining user preferences or reasoning about commonly occurring orderings. This method, while non-intrusive, can be inefficient if the provided training data is not suitably diverse. One practical method of overcoming this challenge is to participate in the learning process.

A. Active Learning

In the general case, Active Learning involves a learner generating or selecting examples to be labeled by an oracle. With a means to participate in the instruction process, a learner is capable of increasing the effectiveness of the tutoring process if she can guide an instructor to provide and label useful examples specific to her current internal state [6], [8].

Within the task learning domain, useful examples are those which inform the learner about previously unknown constraints between the components of a task. When learning task structure, active learning provides a mechanism by which a learner can suggest a potential edge from the environment-space graph via a demonstration query, implicitly specifying the originating vertex (current environment state) and explicitly providing a skill label (the query itself). For example, a robot may ask a human chef if the flour can be added after adding butter in a cookie recipe. In doing so, the learner is requesting the oracle to reveal an edge leading from a vertex containing the current state of the food to a nearly identical vertex where the flour has been added. An instructor can respond to this query positively by utilizing the suggested skill and adding flour, or negatively by performing a different skill such as adding sugar instead.

While a negative response does not necessarily indicate that no edge with the queried label exists from the current environment state, it does suggest that it is unlikely. A particular edge that is known or confidently assumed to not exist can be labeled as an *anti-edge* in the environment-space graph, informing the learner that he should assume no transition is possible matching that particular environment state/skill combination. Anti-edges serve an important role within the task learning domain, as queries are a limited resource [5] and should not typically be used to test previously ruled-out transitions.

As individual instructor habits and preferences can diminish task demonstration diversity, active learning provides a mechanism by which a robot can safely participate in the learning process, encouraging the presentation of varied examples. We show that the process of solving the task learning problem can be expedited with the assistance of an intelligent strategy for encouraging demonstration diversity through active learning, derived from the properties of the *action-space graph*.

B. The Action-Space Graph

While the environment-space graph provides an easily followed execution plan for an arbitrarily complex task, it does not immediately convey features helpful for either discerning task structure or informing an active learning strategy to inspire useful demonstrations. To gain easily interpreted features for satisfying these goals, we utilize the *action-space graph*. Contrasted to the environment-centric view of the standard graphical task representation, the action-space graph provides a skill-centric overview of a task.

Given an environment graph $G = \{V, E\}$, we define the action-space graph as a directed multigraph $H = \{W, F\}$ consisting of a unique set of vertices $W = \{\text{label}(e) \mid \forall e \in E,\ \text{label}(e) \notin W\}$ and a set of edges F connecting vertices in W with labels corresponding to prerequisite environment features derived from the postconditions of previously executed skills (Figure 2).

The action-space graph provides several benefits when compared against the environment-space graph in the context of selecting a skill for a demonstration query. Simple features derived from a partially known action-space graph such as skill distance and skill connectivity can be used to beneficially inform an active learning query mechanism.

C. Query Strategies

For realistic training scenarios, it is important for an active learner to utilize query opportunities as a scarce and valuable resource, optimizing the value derived from each chance to interact with the instructor. Posing too many queries can slow down the training process and potentially cause the instructor to reduce the number of demonstrations or even abandon the training entirely.

As the goal in the task learning domain is to achieve maximum diversity of training examples, the value of a particular query becomes less obvious than merely evaluating whether or not a particular skill can be performed immediately given the current environment state. In some cases, it may prove more valuable to inquire about a skill that cannot be utilized immediately from the current state, with the consequence that the instructor may demonstrate a unique path from the current state to the queried skill. We evaluate the effectiveness of three different active learner query strategies: random, distance-based, and connectivity-based.

1) Random Querying: The baseline mechanism that we compare our strategies against is random selection. The random query mechanism selects a skill at random from the set of skills that have not been seen during the current demonstration of the task. This set excludes the set of skills for which direct transitions are known from the current state. Additionally, the random query mechanism made use of anti-edges, avoiding repetitive or uninformative queries.

2) Distance-based Querying: The distance-based query obtains a distance score for each vertex in the action-space graph (transformed from the learner's current environment-space graph), measuring the shortest path to each. This query mechanism selects the closest skill that has neither been executed nor describes an existing transition from the current state in the environment-space graph. Transitions that are eliminated by anti-edges are also ignored. Ties are broken randomly.

3) Connectivity-based Querying: The connectivity-based query utilizes the degree of each vertex in the action-space graph and its edges' constraint properties to rank potential queries. Each inbound edge for a particular vertex is scored inversely proportionally to the number of environmental prerequisites on its label. For our experiment, we set a base score for each edge equal to the total number of possible prerequisites (obtainable by examining the inbound edge constraints for the terminating vertex of an action-space graph). Inbound edges were given scores of [base_score − prerequisite_count], while outbound edges on a vertex were scored at a flat value of half the base score. The score for a vertex is defined as the sum of its inbound and outbound edge scores. These values were chosen to prioritize well connected vertices with minimal prerequisites. As in the distance-based and random queries, skills that had already been executed this iteration are not considered for selection. Anti-edges are also utilized to avoid known negative queries. The resulting demonstration query is chosen as the remaining skill with the highest score. As in the distance-based query, ties are broken by a skill being chosen randomly from the set of best scoring results.

V. Experiment Design

To explore the performance impact that each of the three query strategies has on a robot utilizing active learning to learn the structure of a task, we conducted a pilot study. Participants were instructed to perform construction tasks in front of the robot, with the expectation that the robot will occasionally ask questions about the current activity. The experiment involved each participant performing 42 demonstrations consisting of 14 examples for each of three different tasks. There were four algorithmic conditions, including one passive learning condition that did not query and three active learning conditions: "Random query", "Distance-based query", and "Connectivity-based query". In querying

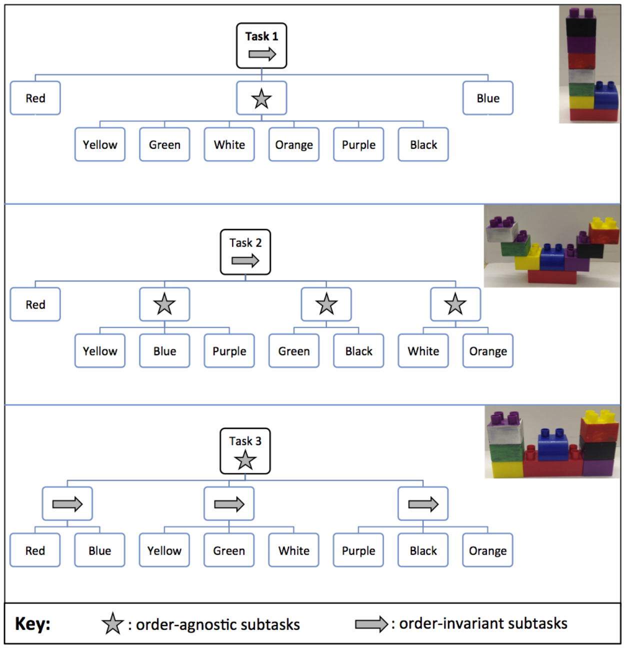

Fig. 3. Construction activity task hierarchies

conditions, the robot only posed open-ended, material-centric demonstration queries. This took the form of a request to use a particular material (building block) next.

For each active learning algorithm, the participant performed each task in succession (task 1, then 2, then 3). Tasks were demonstrated in an interleaved sequence to provide a more naturalistic setting in which to evaluate task instruction, as we seek to investigate a casually instructed robot rather than one with a dedicated instructor expending a concerted effort teaching it. Participants only completed two performances of the task sequence for the "No query" condition, as it was included to familiarize the participant with the experimental procedure. Each of the three active learning conditions involved the participant performing each task four times.

A. Tasks

The first task incorporates a large group of order-agnostic elements within a rigid ordering sequence, identifiable as a chain containing a large clique. Participants were instructed to construct a tower structure (Figure 3-top) with the constraints that the first block placed must be red, the last block placed must be blue, and the order of the other blocks {Yellow, Green, White, Red, Purple, Black} did not matter. This task has 720 valid execution paths. Queries were made after the first, third, and fifth steps of this task.

The second task is an order-invariant sequence of small order-agnostic subtasks, a chain of cliques. Participants were instructed to build a V-like structure (Figure 3-middle) from the bottom up, one row at a time. The rows were (in increasing order): {Red}, {Yellow, Blue, Purple}, {Green, Black}, {White, Orange}. This task has 24 valid execution paths. Queries were made after the first, third, and fifth steps of this task.

The third task consists of an order-agnostic trio of order-invariant subtasks, otherwise described as a clique of chains. Participants were instructed to build three different towers {Yellow, Green, White}, {Red, Blue}, {Purple, Black, Orange} in any order, with the restriction that once a tower is started it must be completed before beginning another (Figure 3-bottom). This task could be completed via six unique execution paths. Queries were made at the beginning of this task, as well as after the second and fourth steps.

B. Procedure

For the duration of the experiment, the participant was seated at the workbench in front of the robot. After being briefed that he would be demonstrating a set of tasks for the robot, the task descriptions were explained to him with the opportunity to ask questions about the tasks to the experimenter. Participants had access throughout the experiment to a description for each task that included a reference image.

Participants were motivated to complete each series of tasks quickly in an effort to encourage natural demonstrations rather than carefully considered training examples. The tablet interface on the table informed participants as to which task they would be performing, as well as when a particular demonstration ended and their structure could be disassembled. Participants were informed that the robot would be testing four different learning algorithms, and that the robot may occasionally ask questions of them.

During a task performance, participants would use a block in their demonstration then press the corresponding button on the tablet to inform the system of the utilized block. Upon making a selection on the tablet, the screen would fade to gray if a query were about to be performed, so the participant would know not to continue until the robot finished its action.

Subsequent to the first demonstration for a each algorithm, the robot would make queries. To perform a query, the robot would move from its resting position and orient itself such that it would be pointing at the block required for the queried skill with its parallel bar gripper. Once the pointing behavior was completed, a synthesized voice asked "Can you use this one next?" before the robot returned to its resting pose. After all robot motion was complete, the tablet screen would display the normal interface and allow the participant to continue. Participants were never given instructions regarding the handling of the robot's inquiries.

Participants were not told any details about the algorithms other than that four were being tested. When the algorithmic condition changed, participants were informed via the tablet interface that the robot would be forgetting everything it had just learned so it could accurately compare methods later.

VI. Results

Six participants (4 female, 2 male) completed the experiment, each providing 42 task demonstrations. Each participant completed the experiment in about 55 minutes. The data set consisting of the learned task graphs trained by

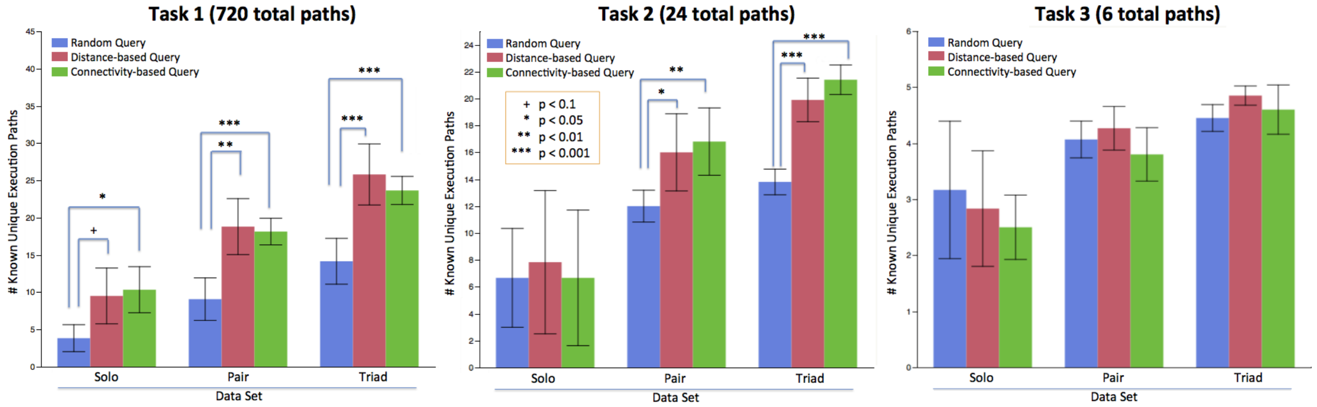

Fig. 4. Active learning strategies compared across tasks and data sets.

individuals is referenced as the *solo* data. A second data set, referenced as the *pair* data, was created from combining task graphs for each possible unique pairing of participants ($\binom{6}{2} = 15$ pairs). A third data set, labeled *triad* data, was created through the combination of every possible unique outcome from selecting three participants ($\binom{6}{3} = 20$ triads). We evaluated our data with the purpose of examining effects that the different active learning strategies have for the three task classes chosen, in the cases of either a single instructor training in isolation or a small group of independently collected, pooled instructor data.

We use the number of known unique paths through the environment-space graph from the origin vertex to the terminal vertex as the primary metric for determining the degree of task comprehension achieved with each query mechanism. Statistical results across were obtained using one-way ANOVAs on path count (dependent variable) and algorithmic condition (independent variable), separated by task and number of instructors (figure 4).

A. Individual Training

For the single instructor demonstration data (N=6, total demonstrations per task=4), only task 1 yielded significant differences between query strategies. We conducted a one-way ANOVA with number of known unique task execution solutions (paths through the SMDP) as the dependent variable and query strategy as the independent variable for each task (1,2,3) and instructor condition (single, pairs, triad). For the single instructor case of the first task, our analysis shows a significant effect of query strategy (F(2,10) = 11.508, $p < 0.01$). Post-hoc tests with a Bonferroni adjustment indicate a significant difference between the random and connectivity-based algorithms (6.5, $p < 0.05$), as well as a marginally significant difference between the random and distance-based algorithms (5.667, p = 0.055). No significant difference was found between algorithms for tasks 2 and 3 in the single instructor scenario.

B. Joint Training

We investigated the effects of combining instructors' data to measure the utility of the feature-based query mechanisms on multi-instructor scenarios. We are particularly interested in this, as we seek to characterize the demonstration diversity that can be obtained from pooling instructors together with respect to the active learning strategy used. Our pilot study results show that the feature-based query strategies encourage diversity across teaching styles, leveraging each instructor's a priori task execution preferences to prompt novel demonstrations. In section 7, we show that this diversity gained from pooling instructors can be more beneficial for learning task constraints than using a single instructor with the same amount of demonstrations.

In this collaborative training scenario, feature-based query strategies perform substantially better than the random query condition. Most importantly, the distance-based query learner completely learned the structure of the second task and the connectivity-based strategy learned the second and third task entirely. This result is motivating, as it suggests that particular query strategies can greatly amplify the benefits of joining different teaching styles. We proceeded to analyze this impact by creating and comparing the pair and triad data set for each task and querying algorithm.

In the pairs data (N=15, total demonstrations per task=8), we find a significant effect of algorithm type for task 1 (F(2,28) = 17.381, $p < 0.001$) and task 2 (F(2,28) = 9.591, p = 0.001). Post-hoc tests on the results for task 1 indicate significant differences between random and connectivity-based algorithms (9.1, $p < 0.001$) and between random and distance-based algorithms (9.7, $p < 0.01$). Post-hoc tests on the results for task 2 indicate a significant difference between both the random and connectivity-based algorithms (4.8, $p < 0.01$) and between the random and distance-based algorithms (4.0, $p < 0.05$). In the second task, the joint instruction process produced a training graph that has a full understanding of the task structure after only 8 demonstrations spread across two non-interacting instructors (4 demonstrations each). The overall performance increases in the third task, but it is unclear that the improvement had any connection to the querying process rather than merely the higher demonstration count, likely due to the low number of unique paths in the graph (6 total).

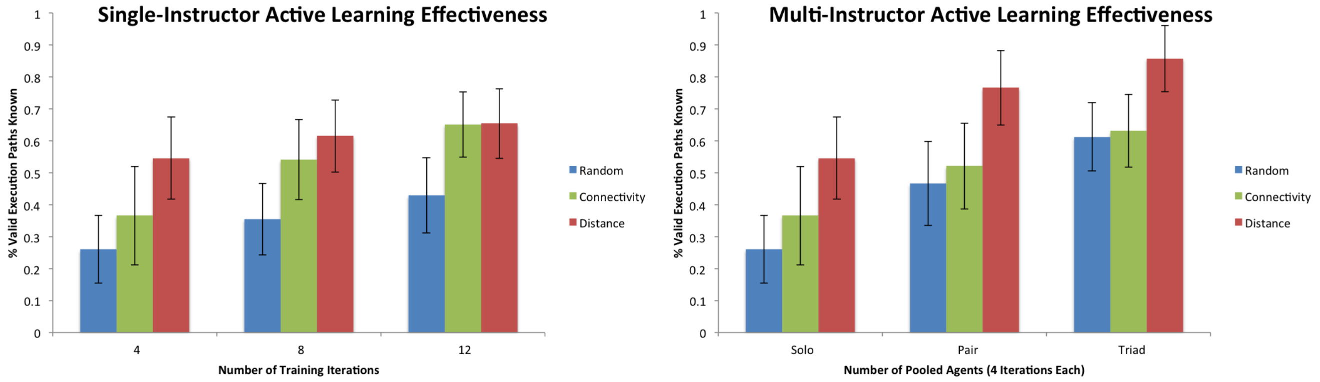

Fig. 5. Results for simulated active learning strategies, averaged over a set of 12 kinesthetically trained food preparation tasks.

Algorithm choice had a significant effect in the instructor triad data (N=20, total demonstrations=12) for task 1 (F(2,38) = 25.107, $p < 0.001$) and task 2 (F(2,38) = 85.597, $p < 0.001$). In the first task, significant differences were again found between random and the other two algorithms: (9.5, $p < 0.001$) for connectivity-based and (11.65, $p < 0.001$) for distance-based. In the second task, there were significant differences found between random and the other two algorithms: (7.6, $p < 0.001$) for connectivity-based and (6.1, $p < 0.001$) for distance-based.

VII. Further Analysis

The results of our first experiment suggested that dividing task demonstrations across multiple instructors may be ideal for maximizing training diversity. In this section, we not only seek to validate our previous results with real-world tasks, but also seek to investigate more directly comparable scenarios where training demonstration counts are kept constant but the number of instructors are varied across query algorithms. For our second experiment, we evaluated the performance of our query generation algorithms on learning a set of 12 real-world inspired tasks. These tasks, from the food preparation domain, contained more steps and more structural complexity than the construction activities from the pilot study, containing 10 to 14 skills and 12 to 24 unique execution paths each.

To perform this analysis, we modeled the pilot study participants' behaviors regarding preferred demonstration sequence, tendency towards unprompted demonstration deviations, and responses to robot queries for each task type and built a simulation environment. An analysis across participant data showed queries that could be accommodated within 3 steps were heeded by the instructor, adjusting her demonstration to cover the queried skill at the earliest allowable time. Queries concerning skills beyond that time horizon were disregarded and had no demonstrated effect on instruction. Participants were likely to remain true to their initial demonstration sequence for each task across algorithms, though their choice of initial sequence appeared to be motivated by the distance of blocks from them on the table (subject to small random modifications for the blocks furthest from the user). In our simulator, we assign each instructor a preferred demonstration path with small random perturbation based upon a fixed arbitrarily chosen canonical base path. Instructor data was pooled in the same manner as the pilot experiment: after all trials were completed, the union of the resulting learner graphs was evaluated.

We simulated 50 trials of learning each task with conditions identical to the pilot study (4 demonstrations per task, 3 queries allowed per demonstration, no queries during the initial demonstration). The simulation involved the production of valid action orderings by virtual instructors, influenced by queries posed from the simulated robot agent. The robot's learned task graph was constructed with a procedure identical to that used in the live experiment. We compare results between querying strategies in the multi-instructor and solo instructor scenarios, presenting the mean percentage of task comprehension in figure 5. In the single instructor scenario, we see average task comprehension rates of 25.9%, 35.4%, and 42.9% for the random query strategy at iterations 4, 8, and 12, respectively. The connectivity-based query generator posted average performances of 36.5%, 54.1%, and 65%, while the distance-based learner achieved comprehension rates of 54.6%, 61.5%, and 65.4%. For the multi-instructor scenario, the random learners received demonstrations for 25.9%, 46.6%, and 61.2% of the possible solutions in the graph. The connectivity-based learners were taught 36.6%, 52%, and 63.1% of graph solutions. Most effective in this experiment, the distance-based algorithm produced average comprehension rates of 54.6%, 76.6%, and 85.8% across all tasks.

These results further highlight the result that these querying strategies are not only effective, but effective in different ways for each instructor. This is evidenced by the increased task network coverage achieved by pooling groups of instructors together. Had the query strategies converged instructors to the same teaching strategy, the pooled data would not show an increase in diversity when compared against the single instructor scenario (with identical total demonstration count). These results also support the justification of using a multi-instructor setup over a single instructor, even when provided the same total training time, as the teaching style of each instructor is expanded in non-convergent ways to cover more of the task graph.

VIII. Discussion and Conclusion

Our results show that feature-based query strategies derived from action-space graphs are a promising avenue to explore within the task learning domain, especially under the restriction of limited data collection. Task structure plays a large role in determining the potential effectiveness of a query, but did not differentiate between the two tested methods enough to suggest a clear winner in our pilot study. In our simulated validation on tasks derived from real world activities, the distance-based querying mechanism showed clear superiority in the multi-instructor domain.

We have utilized a graph representation for SMDPs called the *action-space graph* to provide a set of simple features that can be used to inform an active learner's query generation, obtaining more statistically effective training data from a human instructor than a traditional random exploration strategy. We performed an analysis on the effectiveness of combined instruction from our pilot study, looking at the effects of creating pooled data from small teams of instructors. This result was validated via a simulation of instructors based on a model inspired by our pilot study participants, using a set of 12 complex tasks derived from real-world activities.

Our simulated results reinforce the notion that, when seeking to learn the constraint network of a task, utilizing multiple instructors provides an implicit demonstration diversity benefit over the single instructor case, even given the same total demonstration quantity. This diversity of user preference enables a robot to more quickly learn valid skill sequences by working with each instructors' unique teaching style to produce a wider range of demonstrations.

By increasing the effectiveness of training demonstrations, we provide a way to reduce the human-oriented expense of providing a robot with training data for a given task. The query strategies utilized encouraged instructors to diversify their examples significantly more, the benefits scaling well with the number of instructors.

Our results suggest that learning complex task network constraints from limited quantities of demonstrations is feasible through small-scale collaboration when an effective active learning query strategy is used. These contributions are particularly applicable to collaborative robots that learn from demonstration, helping to remove some of the barriers to achieving task proficiency that have rendered more autonomous learning techniques infeasible.

Acknowledgment

This material is based upon work supported by Office of Naval Research grant #N00014-12-1-0822 ("Social Hierarchical Learning") and NSF award 1139078.

References

[1] B. Akgun, M. Cakmak, K. Jiang, and A. L. Thomaz. Keyframe-based learning from demonstration. *International Journal of Social Robotics*, 2012.

[2] B. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

[3] C. Atkeson and S. Schaal. Robot learning from demonstration. In *International Conference on Machine Learning*, pages 11–73, 1997.

[4] C. Breazeal, C. D. Kidd, A. L. Thomaz, G. Hoffman, and M. Berlin. Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 708–713. IEEE, 2005.

[5] M. Cakmak, C. Chao, and A. L. Thomaz. Designing interactions for robot active learners. *IEEE Transactions on Autonomous Mental Development*, 2(2):108–118, 2010.

[6] M. Cakmak, N. DePalma, R. I. Arriaga, and A. L. Thomaz. Exploiting social partners in robot learning. *Autonomous Robots*, 29(3-4):309–329, 2010.

[7] M. Cakmak and A. Thomaz. Designing robot learners that ask good questions. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 17–24. ACM, 2012.

[8] C. Chao, M. Cakmak, and A. L. Thomaz. Transparent active learning for robots. In *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*, pages 317–324. IEEE, 2010.

[9] L. Cobo, C. Isbell Jr, and A. Thomaz. Automatic task decomposition and state abstraction from demonstration. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 483–490, 2012.

[10] L. C. Cobo, C. L. Isbell, and A. L. Thomaz. Object focused q-learning for autonomous agents. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*, pages 1061–1068, 2013.

[11] A. Dragan, K. Lee, and S. Srinivasa. Legibility and predictability of robot motion. In *Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction*, March 2013.

[12] B. Hayes and B. Scassellati. Challenges in shared-environment human-robot collaboration. In *Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI 2013) Workshop on Collaborative Manipulation*, 2013.

[13] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4, 1996.

[14] B. Kim, C. M. Chacha, and J. Shah. Inferring robot task plans from human team meetings: A generative modeling approach with logic-based prior. 2013.

[15] G. Konidaris, L. Kaelbling, and T. Lozano-Perez. Symbol acquisition for task-level planning. In *The AAAI 2013 Workshop on Learning Rich Representations from Low-Level Sensors*, 2013.

[16] Y. Kuniyoshi, M. Inaba, and H. Inoue. Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *Robotics and Automation, IEEE Transactions on*, 10(6):799–822, 1994.

[17] D. S. Nau, T.-C. Au, O. Ilghami, U. Kuter, J. W. Murdock, D. Wu, and F. Yaman. Shop2: An htn planning system. *J. Artif. Intell. Res.(JAIR)*, 20:379–404, 2003.

[18] S. Nikolaidis and J. Shah. Human-robot cross-training: Computational formulation, modeling and evaluation of a human team training strategy. In *Proceedings of the 8th International Conference on Human-robot Interaction*, 2013.

[19] O. Obst. Using a planner for coordination of multiagent team behavior. In *Programming Multi-Agent Systems*, pages 90–100. Springer, 2006.

[20] R. E. Parr. *Hierarchical control and learning for Markov decision processes*. PhD thesis, University of California, Berkeley, 1998.

[21] J. Shah. *Fluid coordination of human-robot teams*. PhD thesis, Massachusetts Institute of Technology, 2011.

[22] J. Shah and C. Breazeal. An empirical analysis of team coordination behaviors and action planning with application to human–robot teaming. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 52(2):234–245, 2010.

[23] J. Shah, J. Wiken, B. Williams, and C. Breazeal. Improved human-robot team performance using chaski, a human-inspired plan execution system. In *Proceedings of the 6th international conference on Human-robot interaction*, pages 29–36. ACM, 2011.

[24] R. Sutton, D. Precup, S. Singh, et al. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1):181–211, 1999.