Towards Intelligent Arbitration of Diverse Active Learning Queries

Kalesha Bullard[1], Andrea L. Thomaz[2], and Sonia Chernova[1]

***Abstract*— Active learning literature has explored the selection of optimal queries by a learning agent with respect to given criteria, but prior work in classification has focused only on obtaining labels for queried samples. In contrast, proficient learners, like humans, integrate multiple forms of information during learning. This work seeks to enable an active learner to reason about multiple query types concurrently, aimed at soliciting both *instance* and *feature* information from the teacher, and to autonomously arbitrate between queries of different types. We contribute the design of rule-based and decision-theoretic arbitration strategies and evaluate all against baselines of more traditional passive and active learning. Our findings show that all arbitration strategies lead to more efficient learning, compared to the baselines. Moreover, given a dynamically changing environment and constrained questioning budget (typical in human settings), the decision-theoretic strategy statistically outperforms all other methods since it reasons about *both* what query to make *and* when to make a query, in order to most effectively utilize its questioning budget.**

I. Introduction

The paradigm of Learning from Demonstration (LfD) enables an agent to learn a new task from examples provided by a human teacher [1]. In LfD however, the model learned depends on the ability of the teacher to provide appropriate examples to the learner. Placing the primary burden of conveying maximally informative input on the teacher presents an inherent challenge, as it is not feasible to expect every human with task domain knowledge to also understand how an agent models the task and be proficient at teaching it. Yet we want to leverage the domain knowledge of *any* user, independent of teaching skills. Therefore we seek to enable a learning agent to characterize its own uncertainty and autonomously solicit information it needs from the teacher to resolve that uncertainty, thus a collaborator in the learning.

Student-driven agent learning has primarily been encompassed by the field of active learning (AL). Using AL techniques, an agent autonomously selects unlabeled training examples, based upon predetermined selection criteria, and queries an oracle for correct labels [2], [3]. In high-level task LfD, related literature has focused on learning an optimal policy for imitating a human demonstrator's behavior [4], [5], symbol grounding [6], [7], and inferring task constraints [8]. Importantly, the previous work has primarily focused on making *one* specific type of AL query towards generalization along that dimension of the task (*e.g.* taking the optimal action in a state). However, there is a wealth of information an agent can acquire from a human's domain knowledge. And proficient learners, like humans, combine information rather than simply focusing on one type of question. This work is the first to contribute algorithms for enabling an AL agent to *arbitrate* between diverse types of queries, with the goal of autonomously gathering *both* informative features *and* representative instances from the human teacher.

In this work, AL is used to solve a task-situated symbol grounding problem. Symbol grounding is the problem of mapping symbolic representations (labels, concepts) to constructs in the physical world [9]. Assuming no prior knowledge, the robotic learning agent is given a task (*e.g.* serving pasta) and with it, task-relevant concepts (*e.g.* cooking pot, pasta sauce) it must ground, in order to later perform the task in the situated environment. The agent learns to ground the concepts by actively querying its human partner.

The primary contributions of the work are (1) investigating whether enabling a learning agent with strategies for arbitrating between diverse types of AL queries improves learning performance, (2) exploring the design of rule-based (RB) and decision-theoretic (DT) arbitration strategies that enable the agent to acquire and appropriately prioritize feature and instance information useful for the given task, and (3) analyzing the tradeoffs between rule-based and decision-theoretic strategies with respect to learning performance in the agent's situated environment. We conducted an experiment comparing five query arbitration strategies, each gathering both feature and instance information by employing multiple query types, against two baseline approaches for making requests that each only obtain training instances by employing queries of one type. The evaluation was conducted on two tasks consisting of different computer vision datasets. Our findings showed that all RB and DT strategies outperformed both baselines on both tasks. We also found the DT strategy was able to consistently perform at least as well as all RB strategies but had an advantage in that it could additionally reason about *when* to make queries. Thus in the task where environmental change was both more gradual and substantial, similar to many real-world environments, the DT strategy statistically significantly outperformed all other strategies by its ability to adapt to the rate of environmental change and distribute its questions over time, thereby minimizing uninformative requests and acquiring a more representative training sample than any other strategy.

II. Related Work

As motivated in the introduction, we are interested in a *self-driven* learning agent who can leverage the expertise of its human partner in order to acquire the information it needs, by asking questions. Within machine learning literature, this is primarily addressed by AL. Our work is inspired by the scenario of a robot assistant able to acquire groundings necessary for later performing a task in the situated environment. For task learning, AL can enable a robot to both

*This work was supported by the NSF NRI grant.

[1] School of Interactive Computing, Georgia Institute of Technology, Atlanta, Georgia 30332-0250 Email: ksbullard@gatech.edu, chernova@cc.gatech.edu

[2] Department of Electrical & Computer Engineering, The University of Texas at Austin, Austin, Texas 78701 Email: athomaz@ece.utexas.edu

resolve unintended ambiguities during the learning process and explore unseen parts of the state space, in order to create a more *generalized* task representation. Related work on AL for robots has explored the learning of low-level action controllers [10], [11], [12], an optimal policy towards the end of imitating a human demonstrator's behavior [4], [5], grounding of goal state symbols [7], inferring task sequencing constraints [8], and retrieval of objects by the use of curiosity in human-robot dialog [13]. There has also been related work on strategies for introspective and extrospective detection and communication of the learner's knowledge gaps [14]. All of this previous work however has focused on asking *one* specific type of query towards generalization along that dimension of the task.

More closely aligned work includes the proposal of a framework with three types of embodied queries: *label*, *demonstration*, and *feature* queries. It characterizes the value of each in the context of learning lower-level motion trajectories [15]. Yet this work by Cakmak and Thomaz does not include arbitration between the query types, and the entire framework has not yet been applied to the domain of high-level task learning. Additional work within the robotics community looks at requesting feature information from a user. Rosenthal et. al. [16] recommend feature selection as a specific aspect which should be included when asking a question, in order to provide transparency to the human partner; however it does not include an algorithm for enabling the agent to autonomously reason about when to request feature information. Bullard et. al. [17] compare five different approaches for eliciting informative feature subsets from a human teacher and provide insights about the most effective ways to request features from the teacher. Though we can leverage insights from the findings of both, the contribution of this work is in *arbitrating* between several types of AL queries within one coherent questioning framework, such that the learning agent is able to reason about both employing multiple types of questions and acquiring diverse types of information from its human partner.

III. Problem Formulation and Overview

In our problem formulation, the AL agent must solve a *task-situated* symbol grounding problem, defined by [18], in which it must map abstract object symbols to perceptual input associated with physical entities in the agent's environment. Given a set of objects O from a scene in the agent's purview, each object $o \in O$ is represented by a feature vector $\mathbf{x} =< f_1 ... f_m >$. Objects are modeled by the superset of features F extracted from the robot's sensors (*e.g.* color, height). A set of binary classifiers, one for each symbol $y \in Y$, the set of object symbols, each take as input an instance $\mathbf{x}$ and produce a degree of confidence $p(y|\mathbf{x}) = [0,1]$ that $\mathbf{x}$ has label y. For each symbol to be learned, a binary Gaussian process classifier with a radial basis function kernel is trained. This representation was selected because of its efficacy in producing probabilistic predictions of unlabeled instances given only sparse training data.

A. Query Types

To acquire input data, the learning agent must query the human teacher. We utilize three types of candidate queries, which map to two different types of input data to be processed by the symbol grounding models: (1) instances and (2) features. Figure 1 illustrates what type of input data each AL query type provides.

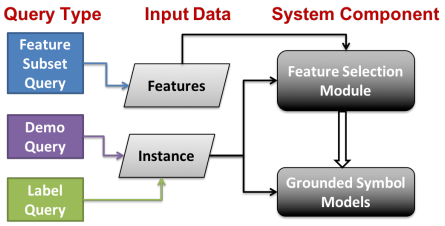


Fig. 1: High-level system diagram mapping query types to type of input each provides and system modules processing the data.

1) Demonstration Queries: Demonstration (or demo) queries (DQ), analogous to active class selection [19], involve the learner requesting a new demonstration of how a concept (symbol) is embodied in the physical world. DQs provide new instances to the system with each demonstration selected by the teacher; the learner is responsible for communicating which symbol it requires a demonstration of.

2) Label Queries: Label queries (LQ), are synonymous to membership or instance queries extensively explored in AL literature [20], [3], [6], [7], [21]. The learner selects an unlabeled instance, based upon predefined selection criteria, and requests the correct label from the teacher. LQs provide new instances to the system as well, but the instances have been specifically targeted and selected by the learner.

3) Feature Subset Queries: Feature subset queries (FSQ) involve the learner requesting a subset of features useful for discriminating between the task-relevant classes; the teacher selects the features. We employ the human feature selection (HFS) approach introduced in prior work and found to be most effective in eliciting feature subsets from humans [17].

B. Learning Episode

The agent uses a single questioning strategy $\hat{s}$ throughout the entire learning episode. The episode begins with the teacher specifying the task and all relevant symbols to be grounded. At each turn t in the episode, the learner observes the state of the world, extracting perceptual input for all objects in the scene. The set of candidate queries consists of (a) one DQ associated with each symbol classifier, (b) one LQ for each object in the scene, and (c) one FSQ, which we constrain to a one-time query, since the answer is not expected to change over time. The learner uses $\hat{s}$ to select a query to make at t, makes the query, then receives teacher feedback. If it receives a new instance, the instance is added to the training set of every symbol classifier (either as a positive or negative example). If provided with a subset of features, it updates all classifiers $y \in Y$ to only consider human selected features from that point in the episode. Lastly, the agent checks stopping criteria to determine whether to continue or end the episode.

IV. Question-Asking Strategies

Our goal was to assess the impact of three separate aspects on the agent's ability to learn the task-relevant concepts: (1) the ability to acquire diverse types of information, (2) the

assignment of priorities to the query types, and (3) the ability to determine when to ask questions. Towards that end, we explore (1) two baseline strategies each making queries of one type, (2) random selection between queries of diverse types, and (3) two categories of *experimental* strategies for arbitration (rule-based and decision-theoretic).

A. Baseline Query Selection

Each *baseline* strategy, employs one query type and acquires only training instances from the teacher, as is typical in interactive learning. Neither has the ability to explicitly reason about when to make queries; they simply acquire data at every turn, until the learning episode concludes.

1. **BL: passive (P)** – Employing only DQs essentially reduces to the traditional LfD scenario, or passive learning, whereby the teacher continually selects examples and the learner passively observes.
2. **BL: active (A)** – Employing only LQs essentially reduces to traditional AL, whereby the agent continually selects unlabeled instances according to predefined criteria (*e.g.* uncertainty) and the teacher provides requested labels.

Passive learning is simulated using stratified random sampling from a generated task dataset. Uncertainty sampling [20], [3] is used for the active baseline, measured as entropy H of object instance $\mathbf{x}$ in the scene. It is computed by:

$$H(y|x) = -\sum_{y \in Y} P_\theta(y|\mathbf{x}) log P_\theta(y|\mathbf{x}) \quad (1)$$

B. Arbitration Strategies

The simplest arbitration strategy we employ is **random query selection (R)**. Given candidate queries of all types, R simply randomly selects one, at each turn. After a feature subset has been requested using HFS, it will select between only DQs and LQs. We validate this strategy to differentiate the relative benefits of multiple query types from the ordering effects of the *experimental* arbitration strategies.

Our experimental approach is inspired by Dialog Management literature, which has traditionally employed rule-based and data-driven approaches for action selection. In keeping with this, we explore two different classes of algorithms for the design of the experimental arbitration strategies.

Experimental: Rule-Based Arbitration

The prioritization of query types for RB strategies follows from machine learning literature. The number of training examples required to learn an accurate model of a concept increases exponentially with the number of features in the state space representation. Thus, machine learning systems typically employ feature extraction as a preprocessing step before training ensues, to increase sample efficiency. Additionally, in AL systems, some passive learning is often done first to obtain a small unbiased training sample for building initial models of the concepts. Then LQs are made on the remaining unlabeled instances, as selected by learner.

Based on these standard practices, we designed RB prioritization as follows: (1) request for task features using HFS, (2) initial demonstrations given by teacher, (3) label requests made by learner for refinement of initial symbol models. We investigate the following rule-based strategies:

1. **ARB: HFS + passive (HFS+P)** – Imposes constraint that features must first be selected by teacher, then passive learning ensues until termination of episode
2. **ARB: HFS + active (HFS+A)** – Imposes constraint that features must first be selected by teacher, then active learning ensues until termination of episode
3. **ARB: HFS + P + A (All/CD)** – Imposes constraint that features must first be selected by teacher, then a minimal set of demonstrations provided by teacher, then refinement of groundings is done through active learning. This strategy also tries to maintain a uniform class distribution.

Experimental: Decision-Theoretic Arbitration

RB strategies explored provide the agent with a seemingly intuitive set of heuristics for selecting a query type at each turn. However, they do not encapsulate any notion of agent goals. They do not allow the agent to directly compare queries of different types and reason about which most enables learning progress. And they do not allow the agent to reason about *whether* to even make a query. Thus we formulate a **decision-theoretic (DT)** arbitration framework which explicitly models the agent's *learning state*, allows direct comparison of diverse query actions, and encodes agent *learning goals* within a multiattribute objective function.

Let D be the set of training instances acquired by the agent, Y the set of task symbol classifiers, O the set of scene objects, and F_y be set of features used by symbol classifier y. Learning state s is represented as the current estimate of the joint probability distribution between objects and symbols, where each element p_{oy} is the posterior probability that object o is an example of symbol y. The current state s is dependent on both D and F_y for each $y \in Y$. The robotic agent's goal is to sufficiently ground and generalize its model of each $y \in Y$.

With respect to the assessment of learning progress, it is not feasible to assume the agent has access to a labeled test set that it can use to evaluate its current performance. Thus the agent needs a different way to both evaluate a candidate query action a and recognize when no query will help it to make it progress towards its learning goals. The expected utility of each candidate $a \in A$ is computed as a linear combination of two goals: (1) maximization of each symbol classifier's ability to discriminate aptly, and (2) minimization of selection bias in the training sample acquired.

1. *Average Classifier Discriminability (ACD)* – Ascertains ability of symbol classifiers to differentiate between most probable and least probable examples of their class. It employs the function:

$$ACD(s) = \frac{1}{|Y|} \sum_{y \in Y} [p_s(y|o_{max}) - p_s(y|o_{min})]$$

   where p_s represents the probabilistic prediction for y in the current state s, and o_{max} and o_{min} represent the objects in the scene predicted to be the *most* and *least* probable examples of symbol y, respectively. ACD value should increase over the course of the learning episode, indicating that the symbol classifiers are improving in their ability to differentiate between examples in O.
2. *Class Distribution Uniformity (CDU)* – Assesses selection bias in the training sample, resulting from the agent

collecting a sample unrepresentative of the underlying distribution. We assume a uniform distribution of symbol classes should be acquired by the agent so as not to bias it towards any task-relevant object, however the formula can be adapted for a nonuniform distribution as well. CDU employs the following function:

$$CDU(s) = \frac{|D_{y_{min},s}|}{|D_{y_{max},s}|}$$

where $D_{y_{max},s}$ and $D_{y_{min},s}$ are each subsets of the training sample. They represent the subsets of positive examples for y_{max} and y_{min}, the symbols most and least represented in the training sample at state s. This value is maximal when the class distribution is uniform.

Combining the above metrics, utility of s is computed as:

$$U(s) = w_1 ACD(s) + w_2 CDU(s) \quad (2)$$

To select an optimal query action a^*, the agent computes the *expected* utility (EU) of taking each $a \in A$, the set of candidate actions, and takes an argmax. EU is computed as

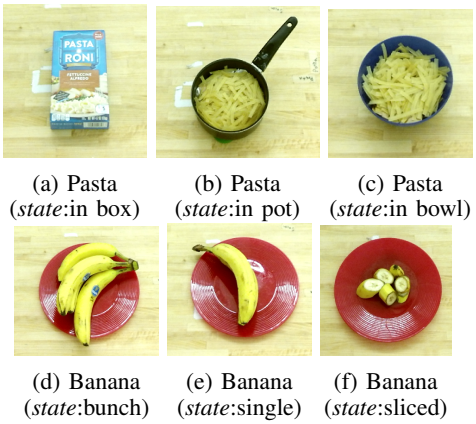
$$EU(a|D,O,Y,F) = \sum_{s'} U(s') * P(s'|a,D,O,Y,F) \quad (3)$$

where s' is a candidate next state resulting from taking a, dependent upon teacher feedback. Importantly, the agent's decision rule is to only take action a^* if $EU(a^*) > U(s)$. Else, the agent makes no query at turn t. Intuitively, this suggests a query should only be made if is expected to improve the state of the learning, *i.e.* engender learning progress.

To assess a DQ for symbol $\hat{y}$, the agent simulates the set of plausible responses by the teacher. Given the agent has requested a demonstration of $\hat{y}$, it assumes the teacher will draw the demonstration from O. Thus $\forall o \in O$, it simulates a resulting state s' by adding labeled instance $< o, \hat{y} >$ to D and computing the $U(s')$ according to Equation 2. We approximate the probability of s' occurring (*i.e.* the teacher providing o as a positive example of $\hat{y}$) as $prob(o|\hat{y})$.

To assess a LQ for object $\hat{o}$, the agent again simulates the set of plausible responses by the teacher. Given the agent has requested a label for $\hat{o}$, it assumes the teacher will provide it a label from Y. Thus $\forall y \in Y$, it simulates a resulting state s' by adding labeled instance $< \hat{o}, y >$ to D and computes the $U(s')$ according to Equation 2. We approximate the probability of state s' occurring (*i.e.* the teacher providing y as a label for $\hat{o}$) as $prob(y|\hat{o})$.

To assess an FSQ, it is too computationally expensive to simulate all feature subsets the teacher could possibly provide, since there are $2^{|F|}$ candidate subsets. Thus, to substantially prune the search space, the agent can use feature subsets outputted by the computational feature selectors of each $y \in Y$. These subsets are the best approximates it currently has for informative features, and prior work has shown that given task features are intuitive, HFS is *at least* as good as computational feature selection [17]. Since we assume task features are semantically interpretable by a human, the agent can expect the feature subset it would receive to be *at least* as good as those outputted by computational methods. Thus, the agent simulates the set of plausible responses by the teacher with the computational feature subsets generated. Utility for an FSQ is computed as follows: $\forall y \in Y$, it simulates a resulting state s' by changing F_y to $F_{y,c}$ for each $y \in Y$, where $F_{y,c}$ is the computational feature subset computed for symbol model y. Thus it retrains all symbol classifiers, given D, but with $F_{y,c}$ as the underlying representation. We approximate the probability of state s' occurring (*i.e.* the teacher providing feature subset $F_{y,c}$) as $1/|Y|$, a uniform distribution over the computational feature subsets generated by the symbol models.



(a) Pasta (*state*:in box) (b) Pasta (*state*:in pot) (c) Pasta (*state*:in bowl)

(d) Banana (*state*:bunch) (e) Banana (*state*:single) (f) Banana (*state*:sliced)

Fig. 2: Illustration of object state changes for *main dish* and *fruit* objects classes in *prepare-lunch* task.

V. Evaluation

Given the research questions being explored, we evaluated three hypotheses: (1) arbitration strategies will outperform baseline strategies since they acquire both informative features and training instances from humans domain experts, (2) prioritizing the acquisition of feature data over instances will result in more efficient learning, and (3) DT arbitration will better adapt within dynamic environments since it additionally reasons about when to make queries.

To evaluate all strategies, we conducted an experiment with two different tasks: (1) a *pack-luncbox* task and (2) a *prepare-lunch* task. Each task uses the same four object symbols: main dish, snack, fruit, and beverage. However, the task datasets have different properties and were created from different image datasets. Each $s \in S$ is evaluated using two metrics: learning accuracy (how well agent identifies unseen examples of each symbol) and sample efficiency (number of questions needed to sufficiently ground all symbols).

A. Data Collection

The pack-lunchbox task assumes all groundings remain static, which means the way the object is embodied in the world does not change. The main dishes (instant noodles) are always packaged, the beverages (water and soda cans) remain bottled or canned, the fruit (apples, oranges, peaches, pears) is whole and ripe, and the snacks (food bags, *e.g.* chips) remain closed. Data for this task was collected from the University of Washington RGB-Dataset of common household objects [22]. The image dataset includes over 200,000 object images in total, encompassing over 300 objects organized into 51 categories (e.g. soda can), with multiple object instances per category (*e.g.* pepsi can, mountain dew can). For each object instance, there are several hundred

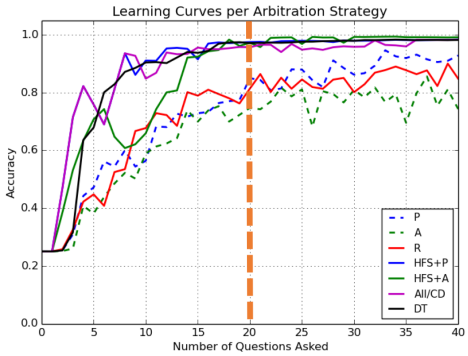
images, captured from three camera viewpoints; a small subset of object instances are additionally captured under different lighting conditions. For the pack-lunchbox task, we only consider object instances relevant to the symbols being grounded. Given images of object instances from the UW dataset, we generated five disjoint training datasets for the pack-lunchbox task and one hold-out test dataset, each training dataset consisting of $n = 3200$ images and the test dataset consisting of $n = 800$ images. All datasets contain images of the same set of task-relevant object instances. For each dataset, stratified random sampling without replacement was used to generate a uniform class distribution.

In the prepare-lunch task, some objects change state, presumably as lunch is being prepared. The motivation for this is even within the same environment, groundings for a particular symbol can change over time (*e.g.* an apple transitioning from whole to sliced, or pasta going from being packaged in a box to being served in a bowl). This property of dynamically changing groundings is an important part of our problem domain and is thus explored in the second task. Towards that end, we varied the states of two objects symbols (main dish and fruit) and allowed the other two to remain static (beverages and snacks). Figure 2 illustrates changes that could reasonably occur as the task is being performed.

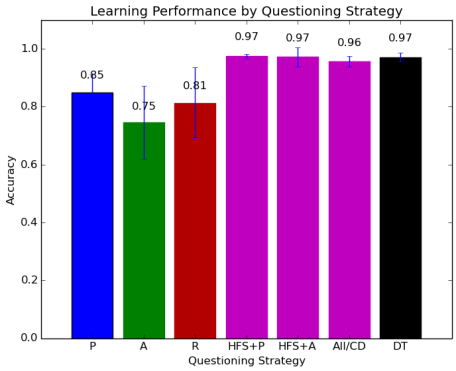
Data collection was done using a Kinect RGB-D sensor on our mobile manipulator robot platform. The sensor is mounted on the head of the robot and was angled to look down at the robot's workspace for taking color and depth images of each object. As data was being collected, the orientation of each object instance was systematically varied, and each object instance was also moved to various positions around the workspace to create different lighting angles. A total of approximately 200 images were taken, four object categories which decompose into 14 different object instances: 3 boxes of pasta, 4 brands of chips, 3 types of fruit, and 4 beverages. From the image dataset created, train and test datasets for the task were generated so as to ensure a representative sample of the object states collected in the data. Given that we aimed for a uniform distribution of classes and the same underlying distribution of object instances in each dataset, not all images were used. The training dataset for the prepare-lunch task contained $n = 80$ images and the test dataset contained $n = 40$ images. For this task, since the number of images of transformed objects taken from our robot was several orders of magnitude smaller than the UW dataset, we seeded each training dataset with the same set of images but added Gaussian noise to the extracted features; we also added Gaussian noise to simulated features for both task datasets. We again generated five training datasets for the second task. For both tasks, the training and test datasets generated are disjoint.

B. Sensory Input

Since our work is intended for a robotic agent, we collected real-world vision data from a robot's camera[1], as well as simulated multi-modal feature information to represent features that would be extracted from other robot sensors[2], resulting in 90 low level features in total. Thus for each object in the scene, F is computed based upon perceptual information taken from multiple robot sensors.



(a) Pack Lunchbox Task



(b) Performance after 20 questions

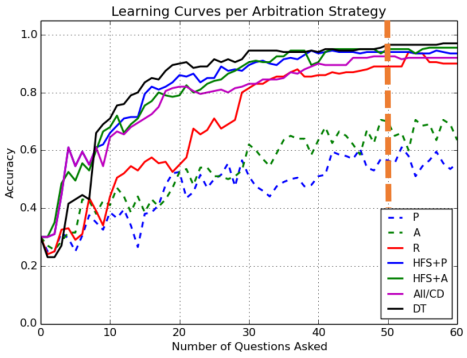
Fig. 3: (a) Accuracy of all strategies for pack-lunchbox task, as a function of number of questions asked. Baseline approaches use computational feature selection; experimental strategies request human-selected features. (b) Comparison of accuracy once learning has stabilized for best strategies (after 20 questions).

C. Experimental Design

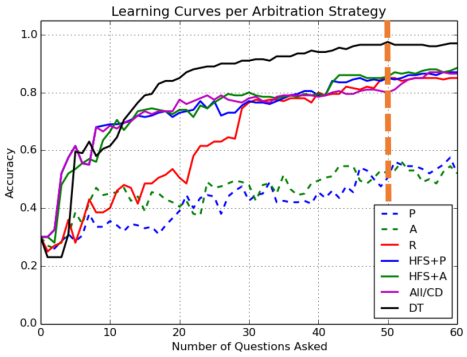
In designing experiments, the agent should be provided with perceptual input that simulates a robotics domain. Thus, we had two goals: (1) since a robot's perceptual system typically outputs one feature vector per cluster in the scene, the system is designed to randomly sample one image per scene object from the specified task dataset each time a new set of observations is generated and (2) since robots typically operate in dynamically changing environments, the system samples a new set of observations every r turns in the learning episode, where r represents the rate of environmental change. For the task where groundings remain static, environmental changes include only viewpoint and/or lighting. Groundings changing over time (Figure 2) means environmental changes may also include physical object state change. At each turn t, O contains only one observation

[1] *object bounding box position, orientation on table, color, size dimensions, area, volume, aspect ratio, visual texture, compactness of object's point cloud, and density of point cloud contour*

[2] *object's location relative to interest points in the environment (e.g. counter top, stove, refrigerator, pantry), the object's location relative to the robot base, absolute location of robot's base in the environment, location of the robot's base with respect to the counter top, the robot's joint positions for each arm, pose of the robot's hands, robot's hand states (open vs closed), weight of the object, and max/min/average volume of noise in the environment over duration of learning episode.*

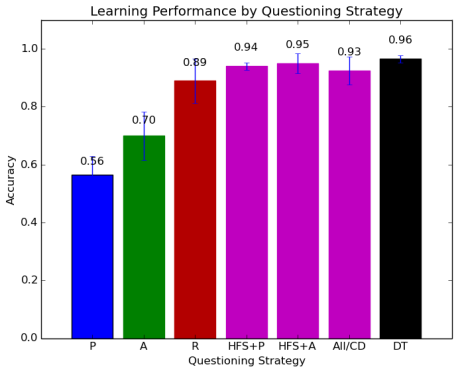


(a) Prepare Lunch Task
(*rapidly* changing environment)

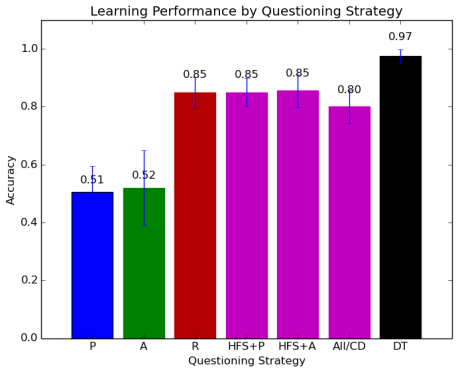


(b) Prepare Lunch Task
(*gradually* changing environment)

Fig. 4: Accuracy of all strategies for prepare-lunch task, as a function of number of questions asked. Baseline approaches use computational feature selection; experimental strategies request human-selected features. Performance under both (a) *rapid* change (every turn) and (b) *gradual* change (every 20 turns).



(a) Performance after 50 questions
(*rapidly* changing environment)



(b) Performance after 50 questions
(*gradually* changing environment)

Fig. 5: Accuracy for prepare-lunch task when learning stabilized for best questioning strategy (after 50 questions), as denoted by the vertical red bars on learning curves. Performance under (a) *rapid* change (every turn) and (b) *gradual* change (every 20 turns)

(image) of each object in the scene. To simulate environmental change, the perceptual system generates a new set of observations. Else, it outputs the set of observations from $t-1$. Given O, the agent decides whether to query, then updates and evaluates all symbol models following feedback given. The teacher for all experiments was one of the authors.

VI. Results

We compare learning accuracy resulting from employing each of the different questioning strategies. To test each strategy $s \in S$, learning accuracy is computed using:

$$E[A_{\mathbb{D}}(s)] \approx \frac{1}{k}\sum_{i=1}^{k}\frac{1}{n}\sum_{\mathbf{x}\in D_i}[1-\delta(h_i^s(\mathbf{x}),y)] \qquad (4)$$

where E is the expected value of the learning accuracy using s on training dataset D_i with respect to distribution $\mathbb{D}$, $h_i^s(x)$ is the hypothesis of the learner using s trained on D_i then tested on instance $\mathbf{x}$ in the test dataset, y is the ground truth label for $\mathbf{x}$, and the quantity $\delta(h_i^s(\mathbf{x}),y)$ is 1 if $h_i^s(\mathbf{x}) \neq y$ and 0 otherwise. Also, n is the number of test instances in each task test dataset and k is the number of task training datasets used. For both tasks, $k=5$; $n=800$ for pack-lunchbox task and $n=40$ for prepare-lunch task.

A. Learning Static Groundings

Figures 3 and 4 show learning accuracy for each $s \in S$ in the pack-lunchbox and prepare-lunch tasks respectively. Each s was given a 40 question budget in the former and 60 question budget in the latter since it is a harder learning problem. We use the Mann-Whitney U-test to compute statistical significance comparisons for each pair of strategies.

Our first hypothesis was that strategies gathering both feature and instance information will outperform baselines acquiring only instances. Our resulting learning curves support this hypothesis for all experimental strategies (*i.e.* rule-based and decision-theoretic), with respect to both learning accuracy and number of questions necessary for learning performance to stabilize. For the pack-lunchbox task (Figure 3), learning performance for all experimental strategies begins to stabilize after approximately 20 questions have been asked, whereas performance does not stabilize for the baseline strategies until approximately 40 questions have been asked. And the baselines still require additional questions to reach the performance of the experimental strategies. Thus on the easier learning task, the experimental strategies are able to sufficiently learn the task-relevant concepts with less than half the number of questions. For the prepare-lunch task (Figure 4), all arbitration strategies significantly outperform both baseline strategies throughout the entire duration of the

learning episodes tested, the random strategy outperforms the baselines for a little more than half of the episode, and the rate of increase for the baseline strategies is very gradual. Thus we do not expect learning performance to stabilize for any of the baseline learners on the more difficult learning task until well after any of the learners using an arbitration strategy conclude their episodes.

The second hypothesis being tested was prioritization of a feature subset request over instance acquisition, imposed by experimental strategies, would result in more efficient learning, as compared to a random arbitrator, which also combines all query types but with no apparent strategy. We found that on average, the random strategy seems to perform on par with the two baseline strategies incorporating only one query type, in the pack-luncbox task, which means it takes much longer to learn the concepts than experimental arbitration strategies for this task. This supports our hypothesis. However, random is able to perform comparably with the experimental arbitration strategies after approximately 30 questions on average, in the prepare-lunch task. Thus all arbitration strategies sufficiently learn the task after approximately 50 questions. This fails to support our hypothesis. To understand why, we examined the episodes more closely. We found that in episodes where the random learner requests human features, learning performance spikes and quickly becomes comparable to that of the experimental strategies thereafter. In episodes where an FSQ is not made, this essentially reduces to the case of randomly selecting between only DQs and LQs; in those cases, we observed learning performance comparable to baseline strategies for the duration of the episode. And since the pack-lunchbox task has over four times the number of objects as the prepare-lunch task (55 vs 12), and thus considers approximately four times the number of candidate queries per turn, R takes substantially longer to randomly select an FSQ in pack-lunchbox than in prepare-lunch. This explains the significant shift in the performance of R in the prepare-lunch task (Fig 4) but not in the pack-luncbox task (Fig 3a); it takes much longer to happen in the latter case. The overall implication is the acquisition of informative features has a significant impact on learning performance; thus if the teacher can provide them, a feature subset request should be prioritized.

B. Learning Groundings that Change over Time

Our final hypothesis was DT arbitration would better adapt within dynamic environments because it additionally reasons about *when* to make queries. We aimed to understand the impact of the *rate* of environmental change on efficacy of arbitration strategy employed. For this analysis, we focus on the prepare-lunch task since we found that rate of environmental change did not noticeably impact learning performance for the pack-lunchbox task, where groundings remain static. In the prepare-lunch task however, the environment *must* change for the agent to encounter all possible symbol groundings, since the groundings themselves change over time. Thus, we use prepare-lunch for exploration of dynamic groundings.

Figures 4a and 4b compare learning performance per number of questions asked in prepare-lunch, given both *rapid* and *gradual* environmental change. When rapid change is occurring (every turn), performance $\forall s \in S$ stabilizes after approximately 50 questions. As shown in Figure 5a, all RB strategies (magenta) and the DT strategy (black) perform comparably. However, under gradually changing conditions (every 20 turns) [3], DT clearly and significantly outperforms all other strategies for most of the learning episode. Figure 5b highlights this by comparing performance of all $s \in S$ at 50 questions, where DT begins stabilizing. Here, all arbitration strategies statistically significantly outperform both baselines. Moreover, DT statistically significantly outperforms *all* other strategies. In all cases, $p < .05$.

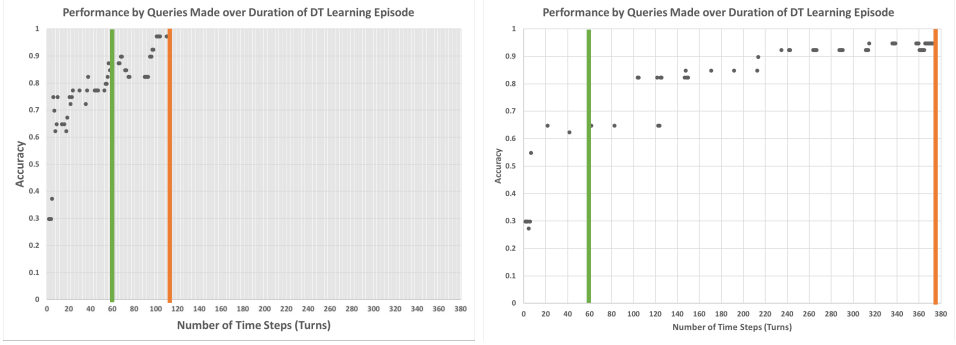
To better understand why DT dominates in this setting, we examine Figure 6, which visually depicts one learning episode for the DT strategy under both rapid and gradual environmental change. Accuracy is plotted as a function of time steps elapsed. Gray vertical bars represent change occurring in the environment. The dots indicate time steps where a query is made. The green vertical bar indicates when all *other* strategies complete their episode (after 60 time steps) since all other $s \in S$ make a query at *every* time step until their questioning budget is depleted. The red vertical bar indicates when the DT agent depletes its questioning budget and completes its episode. Comparing the graphs, under gradual change, the DT agent makes less frequent requests and distributes its questions over 374 time steps. Whereas under rapid change, it takes only 112 time steps to complete its episode and still achieves comparable performance. Even more compelling, under both conditions, its queries are generally made soon after environmental change occurs. Thus illustrating its ability to be adaptive and responsive to environmental change and successfully acquire a representative training sample, independent of the *rate* of change. By comparison, since all other strategies make a query at every time step, they end up acquiring many redundant training examples when the environment is changing slowly. This leads to training samples that inadequately represent the diversity (variance) in each task-relevant class and classifiers that perform sub-optimally on a representative test set. In short, the DT agent allots its questioning budget more wisely, so it is largely unaffected by the slowly changing conditions. In a long-term setting, this is especially compelling because the agent can effectively reason about how to refine its models as a function of change in the environment and does not have to rely on the user to track the state of its knowledge over extended durations or decipher when and how to help the agent update its models.

VII. Discussion

From our experimental investigations, two key insights emerge: (1) enabling the learning agent to ask questions that elicit diverse types of input (*i.e.* both informative features and instances) and appropriately prioritize the query types consistently leads to more efficient learning of the task-relevant concepts and (2) given a dynamic environment and constrained questioning budget (typical in human settings), the DT strategy is able to make the best use of the limited number of questions by deciphering both *when* to make a query and *what* query to make.

The DT strategy is not without its limitations however. DQs are costly to a human teacher because they often

[3] Accompanying video at https://www.kaleshabullard.com/research/



(a) Rapid Environmental Change (perturbation every turn)

(b) Gradual Environmental Change (perturbation every 20 turns)

Fig. 6: Learning Performance as of Number of Time Steps in Learning Episode for *Decision-Theoretic* Strategy

require more effort and time. The DT strategy used an average of 32% and 38% of its questioning budget making demo requests for the pack-lunchbox and prepare-lunch tasks respectively, in the rapidly changing environment, and an average of 52% for the prepare-lunch task in the gradually changing environment. This is substantially more demo requests than the rule-based strategies, which are fixed at 8 DQs, independent of the task, or on average 20% and 13% of the budget for the pack-lunchbox and prepare-lunch tasks respectively. In future work, we are interested in exploring interaction-related attributes in the DT strategy's objective function, so it reasons about an optimal action, considering both learning progress and social aspects of the interaction.

VIII. Conclusion

This work explored the use of rule-based and decision-theoretic strategies for arbitrating between AL queries of different types, to enable a learning agent to acquire diverse types of information (*i.e.* informative features *and* training instances) from a human teacher. We conducted experiments on two different tasks under different environmental conditions, comparing 4 experimental arbitration strategies against baselines of more traditional passive and active learning, as well as random query selection. Overall, the questioning strategies that enabled the learning agent to (1) extract diverse types of information and (2) prioritize acquiring feature information early in the learning episode, more efficiently learned to ground the task-relevant concepts. Moreover, given a dynamically changing environment and constrained questioning budget, the DT strategy was the only strategy able to acquire a representative training sample, independent of the rate of environmental change, because it reasons about both *what* query to make and *when* to query. These findings show that strategic arbitration eliciting diverse types of information from the teacher is able to consistently maximize learning performance when grounding concepts.

References

[1] S. Chernova and A. L. Thomaz, *Robot Learning from Human Demonstration*. Morgan and Claypool Publishers, 2014.

[2] B. Settles, "Active learning literature survey," *University of Wisconsin, Madison*, vol. 52, pp. 55–66, 2010.

[3] Y. Fu, X. Zhu, and B. Li, "A survey on instance selection for active learning," *Knowledge and information systems*, pp. 1–35, 2013.

[4] S. Chernova and M. Veloso, "Interactive policy learning through confidence-based autonomy," *Journal of Artificial Intelligence Research*, vol. 34, no. 1, p. 1, 2009.

[5] M. Lopes, F. Melo, and L. Montesano, "Active learning for reward estimation in inverse reinforcement learning," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2009, pp. 31–46.

[6] C. Chao, M. Cakmak, and A. L. Thomaz, "Transparent active learning for robots," in *ACM/IEEE Int. Conf. on Human-Robot Interaction*, 2010, pp. 317–324.

[7] J. Kulick, M. Toussaint, T. Lang, and M. Lopes, "Active learning for teaching a robot grounded relational symbols," in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. AAAI Press, 2013, pp. 1451–1457.

[8] B. Hayes and B. Scassellati, "Discovering task constraints through observation and active learning," in *2014 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2014, pp. 4442–4449.

[9] S. Harnad, "The symbol grounding problem," *Physica D: Nonlinear Phenomena*, vol. 42, no. 1, pp. 335–346, 1990.

[10] O. Kroemer, R. Detry, J. Piater, and J. Peters, "Combining active learning and reactive control for robot grasping," *Robotics and Autonomous Systems*, vol. 58, no. 9, pp. 1105–1116, 2010.

[11] C. Daniel, M. Viering, J. Metz, O. Kroemer, and J. Peters, "Active reward learning," in *Robotics: Science and Systems*, 2014.

[12] E. Gribovskaya, F. dHalluin, and A. Billard, "An active learning interface for bootstrapping robots generalization abilities in learning from demonstration," in *RSS Workshop Towards Closing the Loop: Active Learning for Robotics*, 2010.

[13] J. Thomason, A. Padmakumar, J. Sinapov, J. Hart, P. Stone, and R. J. Mooney, "Opportunistic active learning for grounding natural language descriptions," in *Conference on Robot Learning*, 2017, pp. 67–76.

[14] M. Majnik, M. Kristan, and D. Skočaj, "Knowledge gap detection for interactive learning of categorical knowledge," 2013.

[15] M. Cakmak and A. L. Thomaz, "Designing robot learners that ask good questions," in *ACM/IEEE Int Conf on Human-Robot Interaction*, 2012, pp. 17–24.

[16] S. Rosenthal, A. K. Dey, and M. Veloso, "How robots' questions affect the accuracy of the human responses," in *IEEE Int. Symp. on Robot and Human Interactive Communication*. IEEE, 2009, pp. 1137–1142.

[17] K. Bullard, S. Chernova, and A. L. Thomaz, "Human-driven feature selection for a robot learning classification tasks from demonstration," in *Robotics and Automation (ICRA), 2018 IEEE International Conference on*. IEEE, 2018.

[18] K. Bullard, B. Akgun, S. Chernova, and A. L. Thomaz, "Grounding action parameters from demonstration," in *IEEE Int. Symp. on Robot and Human Interactive Communication*, 2016, pp. 253–260.

[19] R. Lomasky, C. E. Brodley, M. Aernecke, D. Walt, and M. Friedl, "Active class selection," in *Machine learning: ECML 2007*. Springer, 2007, pp. 640–647.

[20] B. Settles, "Active learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 6, no. 1, pp. 1–114, 2012.

[21] S. Chernova and M. Veloso, "Multi-thresholded approach to demonstration selection for interactive robot learning," in *ACM/IEEE Int. Conf. on Human robot interaction*, 2008, pp. 225–232.

[22] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 1817–1824.