Active Robot Learning for Temporal Task Models

Mattia Racca
Aalto University
Espoo, Finland
mattia.racca@aalto.fi

Ville Kyrki
Aalto University
Espoo, Finland
ville.kyrki@aalto.fi

ABSTRACT

With the goal of having robots learn new skills after deployment, we propose an active learning framework for modelling user preferences about task execution. The proposed approach interactively gathers information by asking questions expressed in natural language. We study the validity and the learning performance of the proposed approach and two of its variants compared to a passive learning strategy. We further investigate the human-robot-interaction nature of the framework conducting a usability study with 18 subjects. The results show that active strategies are applicable for learning preferences in temporal tasks from non-expert users. Furthermore, the results provide insights in the interaction design of active learning robots.

CCS CONCEPTS

• **Computing methodologies** → **Active learning settings**; • **Human-centered computing** → *Natural language interfaces*; *Interaction design*; • **Computer systems organization** → External interfaces for robotics;

KEYWORDS

Interactive machine learning; Human-Robot Interaction

ACM Reference Format:
Mattia Racca and Ville Kyrki. 2018. Active Robot Learning for Temporal Task Models. In *HRI '18: 2018 ACM/IEEE International Conference on Human-Robot Interaction, March 5–8, 2018, Chicago, IL, USA.* ACM, New York, NY, USA, Article 4, 9 pages. https://doi.org/10.1145/3171221.3171241

1 INTRODUCTION

Service robots will be deployed in the future as general assistive devices in dynamic human environments like households, schools and hospitals. In order to be valuable and cost-effective assistants, robots must allow a wide range of customization, especially regarding their skills. Pre-programming robots for every situation is however hardly achievable. Robots need to gain new skills and adapt their behaviour based on the requirements of the environment and the preferences of their users. Even if experts in their domain (e.g. therapists and nurses in hospitals), end users might lack the technical expertise required to shape the robot's behaviours and skills to satisfy their needs. Therefore, robots must be able to learn from people in a natural and effective way.

HRI '18, March 5–8, 2018, Chicago, IL, USA

Most traditional machine learning techniques are not designed to be used in an interactive manner. Interactive Machine Learning (IML) approaches harmonize the presence of people in the learning loop [13]. Several IML paradigms and techniques have been proposed, mainly characterized by the way the human is contributing to the learning. In Learning from Demonstration (LfD) [4, 5], users provide exemplary executions of the desired skill to the learner. In contrast, in Learning from Critique (LfC) the user provides feedback on the robot's learning attempts [3, 10, 22, 35].

Active Learning (AL) approaches enable the learner to query the user (usually referred as *oracle*) to clarify the uncertain elements of the desired skill [2, 15]. AL techniques aim to decrease training time by leveraging the oracle's presence while maintaining the learning performances of passive demonstration-based approaches. AL lowers also the teacher's requirements compared to LfD approaches: users are not expected to always provide informative demonstrations [14, 31] but to reply to the learner's questions targeting the unclear elements.

Although interactive robots can outperform passive ones in both performance and quality of the interaction, a careful design of the Human-Robot interaction (HRI) is needed. Aspects like the transparency of the robot learning process [11, 33], the ability of the user to be a good teacher [9], the timing of the queries or the balance in control over the interaction [7] must be taken into account. Furthermore, efficient ways to mediate between the robot's internal skill representation and the user need to be crafted.

We are interested in studying whether interactive robots can learn complex skills from non-expert users via AL. In this paper, we want robots to learn user preferences about task execution, that are, preferences regarding the temporal order of actions used in a task. The time-related nature of the learning goal creates new challenges, compared to earlier studied classification and concept learning problems [7–9]. First, questions[1] asked by robots need to be in context with the current phase of the demonstrated task. Questions that are out of context might confuse the users and hinder their perception of the robot's capabilities and reliability. Second, queries need to convey information about temporal order and frequencies which are more complex concepts than membership to a group or applicability of a label. Therefore, mediation between the natural language queries and the robot's internal representation of a task is challenging.

We present a query-based AL framework capable of learning user task preferences in an interactive way. While the user provides demonstrations of a task, the robot can ask questions expressed in natural language regarding the observed steps. A template-based query design allows the robot to learn a probabilistic model of the user's preferences whilst being understandable by non-expert

[1] Questions and queries are used as synonyms in this paper.

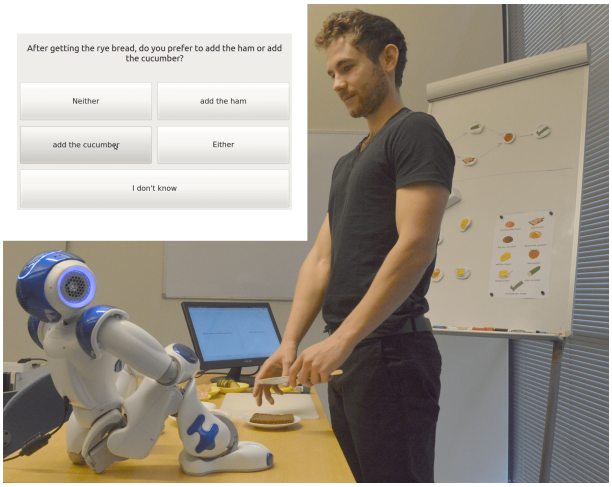


Figure 1: Experimental setup and example of a query, as shown in the touchscreen GUI.

users and in context with respect to the current demonstration. We show in simulations how the proposed framework and two of its variants perform against a passive learning strategy. We then investigate the usability and the HRI aspects of the approach with a user study, where 18 subjects taught a fully autonomous robot the recipes of their favourite sandwiches. We discuss the results in the light of providing future directions for applying more complex AL strategies in robotics.

2 RELATED WORK

In its most traditional definition, Active Learning is a type of semi-supervised learning [2, 15, 34, 36]. While supervised learning strategies constrain the training to a labelled dataset, AL strategies allow the learning agent to interrogate an oracle about unlabelled samples (synthesized by the learner or selected from a pool/stream). If *informative* samples are selected to be labelled (e.g. by choosing samples that maximize the information gain or minimize the predictive uncertainty of the trained model), AL can produce better models faster.

AL's most natural application is the solution of classification problems, with several applications in computer vision [20, 21], text analysis [28] and robotics [7, 16, 17]. Efforts to solve more complex problems like task modelling or policy learning with AL have been done, especially in combination with other IML approaches [12, 18, 24, 35]. Lopes et al. [27] showed how Markov Decision Processes, a popular choice for task modelling, can be learned from ambiguous user feedback with Inverse Reinforcement Learning. They used an AL technique to speed up the shaping of the reward function by requesting feedback on informative samples of the state space. A similar approach was used to learn grounded relational symbols in [24], where an AL strategy generates configurations of objects that are physically achievable by the robotic agent, with the goal of reducing the predictive uncertainty of the model. Hayes and Scassellati [18] applied AL to discover constraints on Task Networks. They enabled the robot to pause the user executing a task and query about the feasibility of the next step (*"Can you do this next?"*). Although able to obtain useful models, these approaches focus their attention on the learning performance rather than their usability and the quality of the interaction, decreasing their value for less experienced or less motivated users. In this paper, we aim for an interactive learning approach which is effective under the performance point of view yet usable by non-expert users.

When using AL approaches, previous work showed that people can develop negative feelings about the role of *label provider* and naturally appreciate participating more in the training process [1, 38]. Richer interaction is, however, challenged by the differences existing between the two agents involved in the learning. Queries must convey the necessary information to enable the learning process and, at the same time, be understandable by the users. As the complexity of the learning task rises, the difficulty of mediating between robot queries and user's understanding of them increases as well. Sadigh et al. [35] presented an example of how user-friendly ways of learning are possible in complex problems. With the goal of encoding user preferences into reward functions guiding the behaviour of autonomous cars, they used an AL comparison approach, presenting informative pairs of synthesised driving trajectories to the users and asking for the preferred one. She et al. [37] proposed a method to teach robots new high-level actions through natural language instructions, simplifying the addition of new skills. Hayes and Shah [19] leveraged the power of natural language to provide robots with ways of explaining their policies, simplifying their use with human collaborators. Following these examples, we propose a template-based query design, translating the robot's learning efforts in questions expressed in natural language.

Given the interactive nature of AL approaches, valuable research examined the HRI aspects of active robots. Rosenthal et. al [33] investigated how the information included in the robot's questions affects the quality of the user's responses, showing that transparent learners help users focus their teaching efforts. Chao et al. [11] investigated the same issue in a concept learning task by tuning the robot's non-verbal behaviours to explain uncertainty about the target concept. Cakmak et al. [7] proposed design guidelines for AL robots by analysing the interaction between non-expert users and robots working together. They studied aspects such as robot transparency, the perceived learning performance and the ease of teaching in the context of a concept learning task. Our work builds on these results, presenting an AL approach using questions expressed in natural language, targeting a more complex learning problem, where the temporal nature of the task influences the quality of the interaction and the query design. With a user study, we investigate the usability of our method, comparing, when possible, our results and observations with the ones in [7].

3 PROPOSED APPROACH

In this section, we first present the model chosen to encode the user's preferences. Second, we present our query-based learning approach.

3.1 Task preference model

The goal of our learning approach is a model of the user preferences regarding the execution of a task. To model the preferred actions and their relative ordering, we choose Markov chains (MCs). In

particular, we model the set of actions A available to the user in a certain task as the states of a MC. MCs can be parametrized as $\theta = \{\pi, T\}$, where $\pi \in \mathbb{R}^{|A|}$ specifies the probabilities of starting the task with each action and the transition matrix $T \in \mathbb{R}^{|A|\times|A|}$ describes the probabilities of transition between actions, where $t_{ij} = p(a_s = a_j \mid a_{s-1} = a_i)$ is the probability of performing action a_j at timestep s knowing that a_i was performed in the previous step.

Knowing their parameters, these models can be used to predict user's future actions based on previously observed ones[2]. Following a LfD strategy, the parameters θ could be learned offline from a batch of demonstrations, defining a demonstration D as a sequence of actions $\{a_1, \ldots, a_n\}$ needed to complete the task.

To increase the interactivity of learning, we learn the MC incrementally, i.e. continuously during the demonstration, by using a Dirichlet-multinomial model over θ [30, Chapter 3]. For each row of the transition matrix T and for the vector π of the starting probabilities, we use an $|A|$-dimensional Dirichlet prior distribution $\mathbf{Dir}(\theta|\boldsymbol{\alpha})$, with $\boldsymbol{\alpha}$ being the model hyperparameters. Besides allowing the incremental learning, the prior distributions over θ give the possibility to an expert user to encode prior knowledge of the task, derived e.g. from common sense or actual task constraints. To compute the posterior distributions, empirical counts (i.e. number of times a_j succeeds a_i and number of times the user starts with a_i) computed from demonstrations are added to the prior hyperparameter values [30]. After the training, the estimated parameters θ are obtained as the means of the posterior distributions.

This LfD strategy has, however, one limitation since it assumes the user to be able to provide not only correct but also informative demonstrations, i.e. covering all the specifications needed to define the model [31]. We target this problem by allowing the robot to ask questions about missing or uncertain details of the task, in addition to observing the user actions.

3.2 Learning by asking questions

Given our modelling choice, we present our query-based learning approach, summarized in Algorithm 1. While a user is giving demonstrations of a task, we enable the robot to learn the model parameters θ passively from the user actions (as the LfD approach presented in Section 3.1) and additionally by asking questions. The robot uses an AL strategy to pick the *best* question to ask at each timestep, given the knowledge encoded in the current model. With the user's answer, the robot then updates its representation. In this section we present the three main components of our approach. First, we present the template-based query design, forming the pool of questions available for the robot. Second, we solve the problem of integrating the information coming from the user back in the learning loop. Finally, we show how our method can pick the most informative questions with a Monte Carlo sampling technique.

3.2.1 Query design. The query design must satisfy conflicting requirements: queries must provide information for the parameter estimation effectively while being understandable by non-expert users, therefore avoiding direct references to probabilities or the underlying learning process. Moreover, the temporal nature of the learning task imposes constraints over the contextualisation of the queries. Out of context questions might not only cause perplexity on the user side, but also distraction from the current task step or inability to reply correctly.

[2]We assume user's actions to be fully observable, either through the user explicitly informing the robot about the current action (as in our experiments) or using reliable sensors to detect them.

Algorithm 1 Proposed Active Learning approach

Input: Action stream $\{a_1, \ldots, a_n\}$, Action set A
Output: User preference model MC

```
while user wants to provide demonstrations do
    initialize a new demonstration D
    while user performs action a and a ≠ end action do
        D ← attach action a to D and passively update model
        q* ← select best question                    [see Algorithm 3]
        r* ← ask the user question q* and wait for the answer
        MC ← update with answer r*                   [see Algorithm 2]
    end while
end while
```

We propose a template based query design with two types of queries: frequency queries (FQs) and disambiguation queries (DQs). As the queries are meant to be used online, i.e. while the user is demonstrating the task, their design takes into account the current action a_s and the previous action a_{s-1} performed.

FQs aim to obtain user's ordering preferences about a pair of actions $\{\mathbf{a_{pre}}, \mathbf{a_{post}}\}$. To convey richer information with the FQs and to avoid the explicit use of probability values, we define a set of frequency adverbs F such as $\{never, sometimes, always\}$. FQs follow the template

FQ: *"After* $\boxed{\mathbf{a_{pre}}}$*, do you* $\boxed{\mathbf{freq}}$ $\boxed{\mathbf{a_{post}}}$*?"*,

where $\mathbf{a_{pre}}$ and $\mathbf{a_{post}}$ are actions and **freq** is a frequency adverb from F. FQs expect a *yes*/*no* answer. We design two subtypes of FQs: FQs about the past (PFQs) and FQs about the future (FFQs). To ground the query in the current step of the demonstration, we limit the choices of the boxed elements in the template ($\mathbf{a_{pre}}$ and $\mathbf{a_{post}}$) to ones related to the current context as described in Table 1. The frequency adverb **freq** can be chosen freely from F.

DQs aim to obtain information about the preferences of the user write respect to a pair of actions $\{\mathbf{a_{choice1}}, \mathbf{a_{choice2}}\}$ following a third one $\{\mathbf{a_{pre}}\}$. DQs follow the template

DQ: *"After* $\boxed{\mathbf{a_{pre}}}$*, do you prefer to* $\boxed{\mathbf{a_{choice1}}}$ *or* $\boxed{\mathbf{a_{choice2}}}$*?"*.

DQs expect the user to reply with one of the following answers: *'Either of these actions'*, '$\mathbf{a_{choice1}}$', '$\mathbf{a_{choice2}}$', *'Neither of these actions'*. Similarly to FQs, we propose two subtypes of DQs: DQs about the past (PDQs),and DQs about the future (FDQs), restricting the choice of the boxed elements as presented in Table 2.

Table 1: Frequency Queries (FQs) template

Type	$\mathbf{a_{pre}}$	$\mathbf{a_{post}}$	Pool size
PFQ	a_{s-1} (previous)	a_s (current)	$\lvert F\rvert$
FFQ	a_s (current)	any from $\{A \setminus a_s\}$	$(\lvert A\rvert - 1) \times \lvert F\rvert$

Table 2: Disambiguation Queries (DQs) template

Type	$\mathbf{a_{pre}}$	$\mathbf{a_{choice1}}$	$\mathbf{a_{choice2}}$	Pool size
PDQ	a_{s-1}	a_s	$\{A \setminus a_{s-1}, a_s, \}$	$\lvert A\rvert - 2$
FDQ	a_s	$\{A \setminus a_s\}$	$\{A \setminus a_s, \mathbf{a_{choice1}}\}$	$(\lvert A\rvert - 1) \times (\lvert A\rvert - 2)$

FQs and DQs aim to estimate the transition matrix T. In order to learn also the starting probabilities π, we additionally propose the starting frequency queries (SFQs) and starting disambiguation queries (SDQs), in the form

SFQ: *"Do you* $\boxed{\textbf{freq}}$ *start with* a_s*?"*
SDQ: *"Do you prefer to start with* a_s *or* $\boxed{\mathbf{a_{choice1}}}$*?"*

and producing, respectively, $|F|$ and $|A| - 1$ possible queries. The templates do not cover repeated actions (i.e. the same action twice or more in a row) as the questions would be clumsy and difficult to answer. The proposed queries form the query pool $\boldsymbol{Q}$, from which the learner will select the most informative query as explained in Section 3.2.3.

3.2.2 Model update. As we use a Dirichlet-multinomial model on the model parameters θ, we need a way to update the robot's knowledge, i.e. to compute the posterior distribution $\mathbf{Dir}(\theta|q, r)$ given the selected query q and the user's reply r. First, we solve the problem of connecting the probabilistic nature of the model and the linguistic nature of the queries and their components. Second, we show how to compute the posterior distribution with a sampling technique.

Borrowing ideas from fuzzy theory [40] and psychology studies on how people perceive frequency adverbs related to probabilities [6, 26], we assign to each $\mathbf{freq} \in F$ a membership function $M_{\mathbf{freq}}(p)$: $S_1 \to [0, 1]$ mapping the probability simplex S_1 into the frequency concept of **freq**. Figure 2e shows a possible choice of membership functions for a set of 3 frequency adverbs. Events with high probability will have high values of membership with frequency adverbs such as *always*. Vice versa, events unlikely to happen (low probability) will score low values of membership with adverbs *always* and *sometimes* and high values of membership with frequency adverbs related to rare events like *never*.

Similarly, we define, for each of the possible answers to the DQs, a membership function $M_{\mathbf{d}}(p(a_1), p(a_2))$: $S_2 \to [0, 1]$ mapping the pair-preference concepts into the probability simplex S_2. Figure 2(a-d) presents a possible choice for the membership functions $M_{\mathbf{d}}$.

The membership functions $M_{\mathbf{freq}}$ and $M_{\mathbf{d}}$ do not have to respect strict constraints regarding their shape and mathematical formulation. On one hand, this increases the generality of our approach: given new query designs, it is enough to encode their concepts in appropriate membership functions. On the other hand, the freedom of choice makes the update of the model parameters, i.e. the computation of the posterior distribution $\mathbf{Dir}(\theta|q, r)$, hardly achievable in closed form.

Algorithm 2 Computation of the posterior distribution over θ given the user answer r to query q

Input: q, r, Prior $\mathbf{Dir}(\cdot|\boldsymbol{\alpha})$, N, $M_{\mathbf{freq}}$ or $M_{\mathbf{d}}$
Output: Posterior $\mathbf{Dir}(\cdot|q, r)$

1: $S \leftarrow$ draw N samples from the prior distribution $\mathbf{Dir}(\cdot|\boldsymbol{\alpha})$
2: $W \leftarrow$ compute weights for S [see Equations 1,2]
3: $\mathbf{Dir}(\cdot|q, r) \leftarrow$ fit a new **Dir** with weighted samples S

To solve this problem, we adopt a sampling based approach, summarized in Algorithm 2. First, we select the prior distribution $\mathbf{Dir}(\cdot|\boldsymbol{\alpha})$ based on the selected query q. If q is a SFQ or a SDQ, we take the prior distribution over π, $\mathbf{Dir}(\pi|\boldsymbol{\alpha})$. For all other types of query, we select the prior distribution over the row of T related to $\mathbf{a_{pre}}$, $\mathbf{Dir}(T_{\mathbf{a_{pre}}}|\boldsymbol{\alpha})$. Second, we draw N samples from the selected prior. For each sample s, we compute a weight w_s, whose value depends on the type of query q, the answer r and the membership functions. For FQs, w_s is computed as

$$w_s(q, r) = \begin{cases} M_{\mathbf{freq}}(s(\mathbf{a_{post}})) & \text{if } r = \text{'yes'} \\ 1 - M_{\mathbf{freq}}(s(\mathbf{a_{post}})) & \text{if } r = \text{'no'} \end{cases} \tag{1}$$

where **freq** and $\mathbf{a_{post}}$ are respectively the frequency adverb and the action used in the query q. If q is a DQ, we compute w_s as

$$w_s(q, r) = M_r(s(\mathbf{a_{choice1}}), s(\mathbf{a_{choice2}})), \tag{2}$$

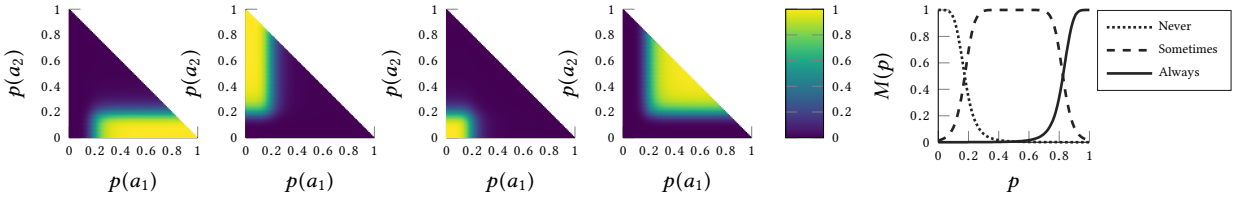
where M_r is the $M_{\mathbf{d}}$ membership function related to answer r while $\mathbf{a_{choice1}}$ and $\mathbf{a_{choice2}}$ are the actions specified in the query q. The membership functions act as filters on the samples S drawn from the prior knowledge, assigning high weight to samples that agree with the user's answer r and low weight to samples in conflict with it. Finally, the posterior distribution $\mathbf{Dir}(\cdot|q, r)$ is obtained by fitting a new Dirichlet distribution on the weighted samples S, using a weighted version of the Expectation Maximization algorithm presented in [29].

3.2.3 Query selection. With the query pool $\boldsymbol{Q}$ designed, the active learner needs a way to select the most informative question to ask in order to speed up the learning. We adapt the idea presented in [34], which selects unlabelled samples to minimize future error rate with a Monte Carlo sampling technique. Instead of unlabelled samples, we select a query q from $\boldsymbol{Q}$ and we integrate the user's reply r in the model. Algorithm 3 summarizes the selection procedure.

As a measure of performance, we use the entropy of the posterior distributions $\mathbf{Dir}(\theta|q, r)$, i.e. the distribution of the model parameter θ given the user's answer r to query q. The entropy of a Dirichlet distribution is always negative and decreases as the distribution becomes more selective. To select which query to choose, we compute the difference of entropy $\Delta\mathbb{H}_q$ in our model before and after each query q in the query pool $\boldsymbol{Q}$. As the user's reply r is unknown at the time of the query choice, we evaluate the *expected* reduction of entropy $\Delta\mathbb{H}_q$ as

$$\begin{aligned} \Delta\mathbb{H}_q &= \overbrace{\mathbb{E}_r[\mathbb{H}(\mathbf{Dir}(\cdot|q, r))]}^{\text{post query}} - \overbrace{\mathbb{H}(\mathbf{Dir}(\cdot|\boldsymbol{\alpha}))}^{\text{pre query}} \\ &= \sum_r p(r|q)\mathbb{H}(\mathbf{Dir}(\cdot|q, r)) - \mathbb{H}(\mathbf{Dir}(\cdot|\boldsymbol{\alpha})), \end{aligned} \tag{3}$$

where $\mathbf{Dir}(\cdot|\boldsymbol{\alpha})$ is the prior related to query q, $\mathbf{Dir}(\cdot|q, r)$ are the posterior distributions (see Algorithm 2) and $p(r|q)$ is the probability of receiving answer r after raising query q. As the probabilities $p(r|q)$ are unknown a priori, we estimate them from the weights



(a) Choice 1. (b) Choice 2. (c) Neither of the actions. (d) Either actions. (e) Frequency functions $M_{\text{freq}}(p)$.

Figure 2: Memberships functions: disambiguation functions $M_{\mathbf{d}}(p(a_1), p(a_2))$ (a-d) and frequency functions $M_{\text{freq}}(p)$ (e).

W of the samples S generated during the posterior computation as

$$p(r|q) = \frac{\sum_i^N w_i(q,r)}{N}, \tag{4}$$

with w_i computed as in Equations 1 and 2. With $\Delta\mathbb{H}_q$ computed for each query $q \in Q$, we can select the *best* query q^*, i.e. the query that is expected to reduce the entropy the most, as

$$q^* = \underset{q}{\text{argmin}}\, \Delta\mathbb{H}_q. \tag{5}$$

Once the user replies with answer r^* to query q^*, the model parameters are updated according to Algorithm 2. After the update, the computed posterior distribution $\mathbf{Dir}(\cdot|q^*, r^*)$ will be used as the new prior for the rest of the learning.

4 SIMULATION EXPERIMENT

We first validated our approach under the learning performance point of view with simulations. We designed a sandwich preparation task with a pool of 9 possible actions; 4 different preference patterns were generated, by varying the required actions and the their ordering, to simulate the preferences of 4 virtual users.

Three AL strategies were compared against a passive learner (**P**), learning only from observing the task execution (LfD approach of Section 3.1). The strategies evaluated were:

- Active Learner (**A**): the proposed AL strategy
- Random Learner (**R**): a variation of **A**, which selects the question to be asked randomly (thus avoiding the query selection presented in Algorithm 3)
- Threshold Learner (**T**): another variation of **A**, which avoids to query the user if the expected entropy reduction $\Delta\mathbb{H}_{q^*}$ is above a threshold value τ.

Algorithm 3 Query selection

Input: User preference model MC, Query pool Q
Output: *Best* query q^*

```
for all q ∈ Q do
    Dir(·|α) ← select prior given query q   [see Section 3.2.2]
    H_prior ← compute entropy of Dir(·|α)
    for all possible answers r to query q do
        Dir(·|q, r) ← compute posterior given q and r [see Algorithm 2]
        H_post(r) ← compute entropy of Dir(·|q, r)
    end for
    ΔH_q ← compute expected entropy reduction for query q [see Equation 3]
end for
q* ← select best query given ΔH   [see Equation 5]
```

Learner **P** was used as a baseline. Learner **R** was studied to investigate whether selecting the best query really improves the learning rate or the contextualization derived by the query design is enough to guide the learning. Learner **T** was chosen to study a cautious learner that queries the user only if worth the interruption caused by the question. We experimentally set $\tau = -0.29$, by analysing the performance of learner **A**. We wanted learner **T** to avoid queries that would reduce the model's entropy by only 50% of the best reduction achieved by learner **A**.

Each learner performed a training session for each virtual user. Each session consisted of 20 demonstrations, following the scheme in Figure 4. All AL strategies used the membership functions shown in Figure 2 and number of Monte Carlo samples $N = 10^5$. The membership functions' shape was chosen and tuned after a few exploratory trials and performance analysis on the simulation. The hyperparameters $\boldsymbol{\alpha}$ were all set to 1, acting as uninformative priors.

To evaluate the true capabilities of each strategy, a single expert user (one of the authors) answered the questions based on the ground truth user preferences. Ground truth models were obtained by generating 200 demonstrations for each of the 4 simulated preference schemes and training a passive learner with them.

4.1 Results

To compare the learning performances, we compute the KL divergence [25] over the parameters θ between the ground truth models and the models obtained by the 4 learning strategies. Figure 3 shows the KL divergence at the end of each demonstration, grouped per strategy and averaged over the 4 preference schemes. As expected, learner **A** and **T** improved their models faster than learner **P** and **R**. After only 5 demonstrations, learner **A**'s and **T**'s models were on average comparable to learner **P**'s model trained with 12 demonstrations. Interestingly, learner **R** produced the worst models. These results show that asking random questions not only slows down the training but also hinders the quality of the model. After 10 demonstrations, learner **A**'s performance stabilized as the same set of questions was repeated without consistent improvements in the model. Learner **T** obtained slightly less accurate models than learner **A** but also asked 59% fewer questions during the first 10

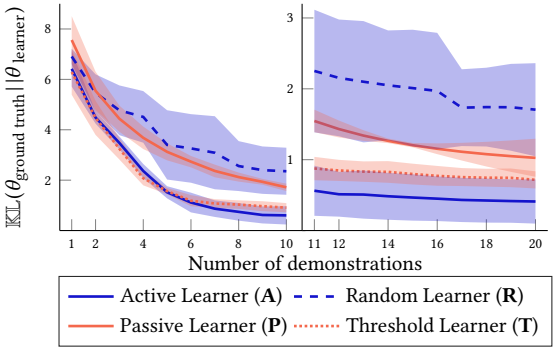


Figure 3: KL divergence on MC parameters θ between the ground truth model and the models obtained with the 4 learning strategies, averaged over 4 training sessions (1 standard deviation highlighted by the shaded area).

demonstrations and 96% fewer questions in the last 10. Computations required by learners **A** and **T** to select the best query among 72 queries took on average around 4.2 ± 0.2 seconds[3]. These results show that the idea of a cautious learner is a good trade-off; however finding the threshold value τ is not trivial and depends on the complexity of the taught task. Running the simulations showed how learners **A** and **T** often selected series of questions of the same type (e.g. FFQs with **freq** = *never* were favoured). This lack of variability in the queries, caused by the minimization nature of the selection process and highly influenced by the choice of parameters, might frustrate or annoy the users and hinder the quality of the training.

5 USABILITY STUDY

To test the usability of the proposed approach, we conducted a study where 18 subjects taught an interactive robot their favourite sandwich recipes. As in the simulation experiment, users could choose between 9 actions to build their sandwiches. A NAO robot named *Nemo* was used as an interactive partner, embodying the 3 AL strategies presented in Section 4. Since previous work had already compared passive versus active strategies [7], we concentrated our analysis on the 3 different flavours of AL.

Experimental setup and robot platform. Figure 1 presents the experimental setup. The NAO robot *Nemo*, sitting on a table where participants were asked to show the recipes, was connected through Robot Operating System (ROS) [32] to an external computer running the framework's software. Nemo used speech synthesis to ask questions and express other utterances. Additionally, Nemo was programmed to look at the subject while asking questions and at the workspace otherwise. To avoid perception errors in the detection of the participants' actions and answers to Nemo's questions, subjects were instructed to declare their performed actions and to answer the questions with a GUI loaded on a touchscreen (connected via ROS). To increase the ecological validity of the experiment, subjects could use real food to prepare the sandwiches and were encouraged to teach Nemo their favourite recipes. Nemo's behaviour was fully autonomous.

[3]Simulation (Matlab code) running on a laptop (Intel Core i7, 8 GB RAM).

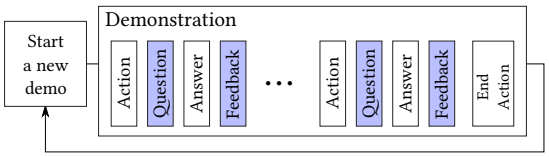


Figure 4: Structure of a learning session. Steps in the blue boxes are performed by the robot. White boxes contain user performed steps.

Conditions and Protocol. During the experiment, each subject interacted with all 3 AL strategies. The order of the strategies was varied and all possible orderings happened 3 times. The parameter values ($M_{\mathbf{freq}}$, $M_{\mathbf{d}}$, τ, N and the prior-hyperparameters $\boldsymbol{\alpha}$) were identical to the simulation. A briefing phase introduced the participants to the robot and the task. Each participant engaged in 3 learning sessions, interleaved by post-session questionnaires. Participants were first asked to design the recipe to be taught to the robot on a whiteboard, provided to help them remember the required steps. No limitations on the complexity of the recipes in terms of number of actions needed or presence of optional steps or multiple choices were enforced. The participants were only required to keep the recipe throughout a session and to provide at least 4 demonstrations. This minimum number of demonstrations was chosen as a trade-off to give time to learner **T** to differentiate itself from **A** while avoiding long and tiring interactions. Experiments lasted on average 45 minutes. Since subjects were free to design the recipes and demonstrated them on average 5.24 times, the learning performance could not be studied as in Section 4.1.

Figure 4 shows the structure of a learning session[4]. First, the user starts a new demonstration by acting on the touchscreen GUI. After each action performed by the user, the robot asked a question q^* (visualized also on the GUI), waited for the user's answer r^* and then updated its model. After each answer, Nemo provided two kinds of feedback utterances based on the user's answer r^*. If the answer was the expected one, i.e. $r^* = \text{argmax}_r\, p(r|q^*)$, the robot used sentences like *"I knew it!", "Okay!", "I was expecting that"*. Otherwise, the robot would reply with surprised utterances like *"Oh really?", "Good to know", "I was not expecting this"*. In the learner **T**'s case, when no question was needed, the robot said either *"I don't have questions on this step, please continue"* or *"This passage is clear, please continue"*. To allow the subjects to express incomprehension towards Nemo's questions, we added an extra answer *"I don't know"*, triggering excuses from the robot (*"I'm sorry that my question was not clear"*) and no model update.

To avoid the lack of variability in the robot's queries observed in simulation, we forced all learners to alternate their choice between the two main query types, FQs and DQs, producing a more natural interaction. Finally, to ensure the grammatical correctness of FQs with the **freq** = *never*, their wording was changed from *"Do you never do. . .?"* to *"You never do . . ., am I right?"*.

Participants. Eighteen participants (age $M = 34$, $SD = 10.4$, female 55%) were recruited in a university campus. Participants had

[4]Video recording of a learning session is available at vimeo.com/mattiaracca/hri18.

diverse education levels (1 high school diploma, 5 bachelor's degrees, 10 master's degrees, 2 doctoral's degrees), with 8 participants having an engineering background and none having hands-on experience with NAO robots. At the end of the experiment, participants were rewarded with a movie ticket (12 $ monetary value).

Questionnaires. After each learning session, subjects filled a post-session questionnaire (inspired by [7]) with the following 1-7 Likert scale questions:

- How well do you think Nemo learnt the recipe (in percent)? (1 - *0%*, 4 - *50%*, 7 - *100%*)
- While showing the recipe, was it clear to you if Nemo was learning the recipe? (1 - *Not clear at all*, 7 - *Extremely clear*)
- Were Nemo's questions bothering or distracting you from your task? (1 - *Extremely distracting*, 7 - *Not bothering at all*)
- How easy was it to teach Nemo the recipe? (1 - *Extremely difficult*, 7 - *Extremely easy*)
- How in context were Nemo's questions with respect to your recipe steps? (1 - *Completely out of context*, 7 - *Extremely in context*)

The questions are presented here in the same order of their respective scores presented in Table 3 and 4. Each question included an optional comment section labelled *"Why? Please explain"*. At the end of all 3 learning sessions, a post-experiment questionnaire collected preferences about the 3 strategies and open suggestions about the interaction design. Subjects could navigate freely the web questionnaire and modify their answers before submission. Besides questionnaires, we logged Nemo's questions, the demonstrations provided by the users (actions declared with the touchscreen) and the answers.

5.1 Results and Discussion

Scores from the questionnaires are summarized in Table 3. We ran Shapiro-Wilcox tests for normality on the scores, rejecting the null hypothesis for all scores and conditions (p<.05). Hence, non-parametric tests were used to compare each condition on the questionnaire scores (Kruskall-Wallis tests) and to pair-wise compare the three conditions (Wilcoxon signed-rank tests). Table 4 reports test statistics and p−values.

Perceived performance and transparency of the learning process. Subjects considered learner **T** as the best performing strategy. However, as we saw from the simulation experiment, learner **T** had slightly worse performances with respect to **A** as asking fewer questions (29% fewer questions than **A** and **R** on average) decreased the amount of information available for training. Interestingly, subjects perceived the decreased number of queries as a sign of improved learning. To support this thesis, subjects commented with e.g. *"Nemo had no questions at the end of the session"* and *"Nemo had less and less questions over time"*. In contrast, the continuous flow of questions of learners **A** and **R** was seen as sign of poor learning. Subjects would comment with *"Nemo kept asking same/similar questions"* (**A**) and *"He kept asking same or rather irrelevant questions even after 5 demos"* (**R**).

Subjects' preference for fewer questions was also seen in the comments under the second question. In particular, subjects experiencing first learner **T** had harsher comments against the learners **A** and **R** (e.g. *"I just don't know if Nemo learned because he didn't tell me"* or *"Nemo did not say that it knows this phase so I was not so sure about its learning progresses"* respectively). Interestingly, very few comments mentioned Nemo's feedbacks, showing that them alone are not enough to ensure transparency. One solution to increase the transparency would be to allow users to query the learner to verify its progresses, as suggested in [1, 23].

Ease of teaching and distracting factors. Subjects perceived teaching learner **T** as easier compared to both the other strategies, while no significant difference was seen between learners **A** and **R**. The discriminant was again the number of questions asked. The comments backing this observation ranged from *"Too many questions"* to *"Questions were good, there were just too many"*. This result does not agree with what was observed in [7]. In their classification task, the fully active strategy was considered the easiest because the constant flow of questions guided the teacher. This suggest how the best strategies might change given the task to solve.

The questions posed by Nemo were in general not seen as too complicated and only 1.6% of them received the *"I don't know"* answer, with no difference between learners. However, learners **A** and **T** often used questions expecting a negative reply from the user. As an example, a FDQ expecting a *"Neither of those"* answer is a powerful (ruling out two options at the same time) but at the same time misleading query, according to some subjects (e.g. *"(Nemo) could have asked simpler questions, without using difficult logic constructions"*). Previous work already observed people's bias for teaching via positive feedback or examples [1, 39] and our observations show how this bias might exists also towards negative answers. A possible solution to this would be a more advanced query selection, taking into account the user's taste regarding questions.

Table 3: Questionnaire scores (1-7) for all AL learning strategies: first quartile (Q1), median, third quartile (Q3). The plot graphically compares the median ratings.

Questionnaire scores	**R**	**A**	**T**
Performance (7=best)	4,**5.5**,6	5,**6**,6	6,**7**,7
Transparency (7=clearest)	3,**5**,6	5,**5**,6	6,**6**,7
Distraction (7=least distracting)	3,**3.5**,5	2,**4**,5	4,**5**,5
Ease of teaching (7=easiest)	2,**4**,5	2,**4**,5	5,**5**,6
Context (7=most in context)	3,**5**,6	4,**6**,6	5,**6**,6

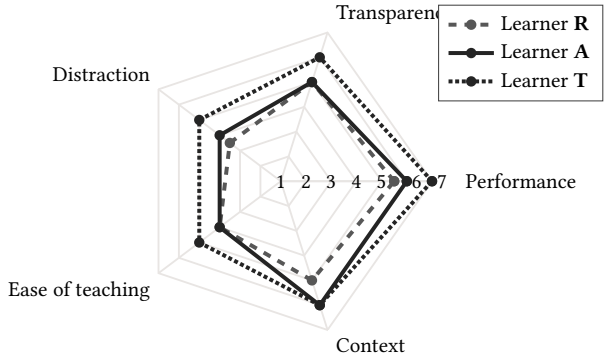


Table 4: Test statistics and $p-$values of multiple-comparison (Kruskall-Wallis) and pair-wise comparison (Wilcoxon signed-rank) across all learning strategies on questionnaire scores.

	Kruskall-Wallis	Random vs Active	Random vs Threshold	Active vs Threshold
Performance	H=10.39, p<.01**	Z=-1.31, p>.05	Z=-2.65, p<.01**	Z=-2.17, p<.05*
Transparency	H=8.64, p<.05*	Z=-1.63, p> .05	Z=-2.21, p<.05*	Z=-1.55, p>.05
Distraction	H=3.42, p>.05	Z=0.28, p>.05	Z=-1.76, p>.05	Z=-1.98, p<.05*
Ease of teaching	H=10.22, p<.01**	Z=-0.26, p>.05	Z=-2.97, p<.01**	Z=-2.89, p<.01**
Context	H=4.31, p>.05	Z=-1.49, p>.05	Z=-2.26, p<.05*	Z=-0.74, p>.05

Finally, the use of the touchscreen by the subjects might have influenced the naturalness of the interaction. We believe that enabling the robot to perceive user's actions and answers would improve the overall quality of the interaction equally for all the studied learning strategies, thus not affecting our conclusions.

Contextuality of questions and their evolution over time. Although questions used by learners **A** and **T** were on average considered more in context than the one from **R**, significant difference was only seen between **T** and **R**. Subjects often commented **R**'s questions to be *random* or *irrelevant*. Comments on learners **A** and **T** were more positive and agreeing with each other: the queries were perceived as *sensible*, *informative* and even *clever*.

From the comments, we could also see how some participants tried to understand Nemo's learning strategies. Subjects realized how learners **A** and **T** improved the quality of their questions as more demonstrations were given, commenting with *"Questions were more relevant after a couple of demonstrations"*. Surprisingly, also learner **R** received comments of the same nature, for example *"At the beginning, (questions were) out of context, then it got better"* or *"The final questions were quite accurate"*. Some fortunate random selections of queries could explain these positive comments. The query design could be another possible reason, mitigating with its contextuality the random choices of learner **R**.

Another interesting point is the way different subjects looked at repetitions in questions. Some subjects saw repeated questions as a sign of poor learning (e.g. *"(Nemo was) repeating the same questions and not learning much"*). Others, however, appreciated these queries, commenting that *"(Questions were) quite in context, Nemo appeared to be confirming things"* or *"(Nemo) seemed to confirm things instead of asking randomly"*. The exploratory nature of the query selection mechanism was also not perceived equally. While some subjects appreciated when Nemo was trying to rule out uncommon options, others saw this as a loss of time and a sign of poor reasoning. Finally, some subjects complained (with low values of context score) about Nemo's lack of common sense and basic knowledge about the task (although this issue is solvable with adequate priors). These contrasting opinions suggest that different strategies might suit different users, based on their preferences and their skill as teachers, as also suggested in [7].

Discussion. From the post-experiment questionnaire, seven subjects expressed preference towards the current interaction design that interleaves demonstrations and questions. However, an equal number of subjects suggested an alternative approach, separating the demonstrations of the task from a separate question phase. We understand the appeal of this option, combining the benefits of AL while avoiding the disliked constant flow of questions. However, we believe that separating the two phases would make the questions in the current design more complicated to answer, forcing the user to remember what happened in the past.

The post-experiment questionnaire confirmed the users' preference toward learner **T** (12 subjects) observed in the post-session questionnaire's scores, followed by learner **A** preferred by 4 subjects and learner **R** with 2 preferences. Subjects' comments show again preferences towards the reduced amount of questions and the perceived faster learning rate of learner **T**, caused by its increased transparency. Given these results, learning strategies that take into account both performances and user's experience like learner **T** should be preferred. The threshold τ used by learner **T** was however set manually in the current implementation. To use this approach at its full potential, principled ways to set τ based on task features (e.g. number of actions available) or on human factors (e.g. tiredness or distraction of the user) need to be developed.

6 CONCLUSIONS

In this work, we proposed an active learning framework for modelling user preferences in temporal tasks through questions expressed in natural language. The framework can pose the most informative questions to learn efficiently and request the user's intervention only when beneficial. A user study showed that participants valued the interactiveness of the active learning strategies, confirming findings from previous research. In particular, robot transparency played a pivotal role in the interaction, impacting the perceived robot learning performance. We also observed how non-expert users may value aspects such as understandability and variability of questions more than their optimality. A limitation of the proposed task model is that it cannot capture complex task constraints and further work is needed to develop active learning for more complex models like MDPs and Task Networks. Query selection techniques taking into account not only the optimality but also user preferences regarding questions show major potential for improving interaction.

Acknowledgements. This research was supported by the Strategic Research Council at the Academy of Finland, decision 292980.

REFERENCES

[1] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *AI Magazine* 35, 4 (2014), 105–120.
[2] Dana Angluin. 1988. Queries and concept learning. *Machine learning* 2, 4 (1988), 319–342.

[3] Brenna D. Argall, Brett Browning, and Manuela Veloso. 2007. Learning by demonstration with critique from a human teacher. In *Human-Robot Interaction (HRI), 2007 2nd ACM/IEEE International Conference on*. IEEE, 57–64.

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

[5] Christopher G. Atkeson and Stefan Schaal. 1997. Robot learning from demonstration. In *Proceedings of the 14th International Conference on Machine Learning (ICML)*. 12–20.

[6] Bernard M. Bass, Wayne F. Cascio, and Edward J. O'connor. 1974. Magnitude estimations of expressions of frequency and amount. *Journal of Applied Psychology* 59, 3 (1974), 313.

[7] Maya Cakmak, Crystal Chao, and Andrea L. Thomaz. 2010. Designing interactions for robot active learners. *IEEE Transactions on Autonomous Mental Development* 2, 2 (2010), 108–118.

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

[9] Maya Cakmak and Andrea L. Thomaz. 2014. Eliciting good teaching from humans for machine learners. *Artificial Intelligence* 217 (2014), 198–215.

[10] Thomas Cederborg, Ishaan Grover, Charles L. Isbell, and Andrea L. Thomaz. 2015. Policy shaping with human teachers. In *Proceedings of IJCAI*. 3366–3372.

[11] Crystal Chao, Maya Cakmak, and Andrea L. Thomaz. 2010. Transparent active learning for robots. In *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*. IEEE, 317–324.

[12] Crystal Chao, Maya Cakmak, and Andrea L. Thomaz. 2011. Towards grounding concepts for transfer in goal learning from demonstration. In *Development and Learning (ICDL), 2011 IEEE International Conference on*, Vol. 2. IEEE, 1–6.

[13] Sonia Chernova and Andrea L. Thomaz. 2014. Robot learning from human teachers. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 8, 3 (2014), 1–121.

[14] Sonia Chernova and Manuela Veloso. 2009. Interactive policy learning through confidence-based autonomy. *Journal of Artificial Intelligence Research* 34, 1 (2009), 1.

[15] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. 1996. Active learning with statistical models. *Journal of artificial intelligence research* 4, 1 (1996), 129–145.

[16] Joachim de Greeff, Frédéric Delaunay, and Tony Belpaeme. 2012. Active robot learning with human tutelage. In *Development and Learning and Epigenetic Robotics (ICDL), 2012 IEEE International Conference on*. IEEE, 1–6.

[17] Victor Gonzalez-Pacheco, Almudena Sanz, Maria Malfaz, and Miguel A. Salichs. 2014. Using novelty detection in HRI: Enabling robots to detect new poses and actively ask for their labels. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*. IEEE, 1110–1115.

[18] Bradley Hayes and Brian Scassellati. 2014. Discovering task constraints through observation and active learning. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 4442–4449.

[19] Bradley Hayes and Julie A. Shah. 2017. Improving Robot Controller Transparency Through Autonomous Policy Explanation. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 303–312.

[20] Steven C.H. Hoi, Rong Jin, Jianke Zhu, and Michael R. Lyu. 2006. Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd international conference on Machine learning*. ACM, 417–424.

[21] Ajay J. Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. 2009. Multi-class active learning for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2372–2379.

[22] Bradley W. Knox, Peter Stone, and Cynthia Breazeal. 2013. Training a robot via human feedback: A case study. In *International Conference on Social Robotics*. Springer, 460–470.

[23] Todd Kulesza, Simone Stumpf, Weng-Keen Wong, Margaret M. Burnett, Stephen Perona, Andrew Ko, and Ian Oberst. 2011. Why-oriented end-user debugging of naive Bayes text classification. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 1, 1 (2011), 2.

[24] Johannes Kulick, Marc Toussaint, Tobias Lang, and Manuel Lopes. 2013. Active Learning for Teaching a Robot Grounded Relational Symbols. In *Proceedings of IJCAI*. 1451–1457.

[25] Solomon Kullback and Richard A. Leibler. 1951. On Information and Sufficiency. *Ann. Math. Statist.* 22, 1 (03 1951), 79–86. https://doi.org/10.1214/aoms/1177729694

[26] Roy S. Lilly. 1968. The qualification of evaluative adjectives by frequency adverbs. *Journal of Verbal Learning and Verbal Behavior* 7, 2 (1968), 333–336.

[27] Manuel Lopes, Thomas Cederborg, and Pierre-Yves Oudeyer. 2011. Simultaneous acquisition of task and feedback models. In *Development and Learning (ICDL), 2011 IEEE International Conference on*, Vol. 2. IEEE, 1–7.

[28] Kachites A. McCallumzy and Kamal Nigamy. 1998. Employing EM and pool-based active learning for text classification. In *Proceedings of the 15th International Conference on Machine Learning (ICML)*. 359–367.

[29] Thomas Minka. 2000. *Estimating a Dirichlet distribution*. Technical report, MIT.

[30] Kevin P. Murphy. 2012. *Machine learning: a probabilistic perspective*. MIT press.

[31] Andrew Y Ng and Stuart J Russell. 2000. Algorithms for inverse reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning (ICML)*. 663–670.

[32] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. 2009. ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*, Vol. 3. Kobe, 5.

[33] Stephanie Rosenthal, Anind K. Dey, and Manuela Veloso. 2009. How robots' questions affect the accuracy of the human responses. In *Robot and Human Interactive Communication, 2009. RO-MAN. The 18th IEEE International Symposium on*. IEEE, 1137–1142.

[34] Nicholas Roy and Andrew Mccallum. 2001. Toward Optimal Active Learning through Sampling Estimation of Error Reduction. In *Proceedings of 18th International Conference on Machine Learning*.

[35] Dorsa Sadigh, Anca Dragan, Shankar S. Sastry, and Sanjit A. Seshia. 2017. Active Preference-Based Learning of Reward Functions. In *Robotics: Science and Systems*.

[36] Burr Settles. 2012. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6, 1 (2012), 1–114.

[37] Lanbo She, Yu Cheng, Joyce Y. Chai, Yunyi Jia, Shaohua Yang, and Ning Xi. 2014. Teaching robots new actions through natural language instructions. In *Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on*. IEEE, 868–873.

[38] Simone Stumpf, Vidya Rajaram, Lida Li, Margaret Burnett, Thomas Dietterich, Erin Sullivan, Russell Drummond, and Jonathan Herlocker. 2007. Toward harnessing user feedback for machine learning. In *Proceedings of the 12th international conference on Intelligent user interfaces*. ACM, 82–91.

[39] Andrea L. Thomaz and Cynthia Breazeal. 2008. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence* 172, 6-7 (2008), 716–737.

[40] Lotfi A. Zadeh. 1965. Fuzzy sets. *Information and control* 8, 3 (1965), 338–353.