# Learning Social Interaction from the Wizard: A Proposal

**W. Bradley Knox, Samuel Spaulding, and Cynthia Breazeal**
Media Lab
Massachusetts Institute of Technology
Cambridge, MA

### Abstract

Learning from demonstration (LfD) is a widely-used technique for programming robot behavior. Despite its flexibility and applicability to a wide range of tasks, LfD is most typically used to learn non-social behaviors. To our knowledge, no prior work has learned robot behavior from demonstration for interaction that is primarily social in nature.

In this paper, we conjecture that LfD can be used to develop robot behavior capable of interacting socially with humans, in scenarios such as a sharing the use of a playful and educational app. These demonstrations, we argue, should be provided within a Wizard-of-Oz paradigm. Thus, we refer to LfD for social human-robot interaction as learning from the wizard (LfW). We describe the corresponding algorithmic and experimental framework and explore challenges in designing LfW systems. Finally, we describe two ongoing projects involving LfD for human-robot interaction, along with a robotic platform and cognitive architecture that are common between the two projects.

## Introduction

Learning from demonstration (LfD) is a common approach for developing robot control. In LfD, demonstrations of correct or desired behavior are provided to a learning algorithm, which attempts to derive a control policy that effectively emulates the demonstrator. Demonstrations are often provided by a live person, though they can also be provided by an algorithm.

Thus far, robot learning from demonstration has been used largely for tasks involving motor-based interaction with the physical environment around the robot. In this paper, we argue for the novel application of learning from demonstration to derive robotic behavior for social interaction. We also argue that the human interaction partner should be made to think that the robot is acting autonomously when actually controlled by a demonstrator, which makes the demonstrations a form of Wizard-of-Oz control. We thus refer to the use of LfD to learn social interaction more concisely as learning from the wizard, or LfW. We distinguish this approach from previous work involving learning behavior for human-robot interaction from demonstration by emphasizing our focus on learning *intrinsically social* behavior. We define intrinsically social behavior here to be actions that primarily function as a means to convey information to another party, not to accomplish some physical task. Prototypical examples include gaze cues, pointing gestures, verbal expressions and facial expressions, among other things.



Because we are focusing on learning intrinsically social behavior, the success of the interaction depends heavily on effective expression of robotic social cues. Accordingly, the robot's space of possible actions will include social cues such as communicative gaze, gestures, and verbal utterances. We note that though the action space must have social dimensions, the sensing space is not necessarily social. Any sensory data that provides sufficient context for effective social action is acceptable. That said, socially-oriented sensory data (e.g., sensing emotions of interactors) might be necessary in practice to provide sufficient context for determining behavior.

We conjecture that learning from demonstration is particularly well-suited for deriving behavioral policies for social interaction with a human. One alternative to LfW is to program behavior directly, for instance via if-then rules, dynamic Bayesian networks, or finite state machines. Another alternative is to specify an evaluation function (e.g. a reward function or fitness function) and allow the robot to autonomously improve its performance according to the evaluation function. LfW appears to have several advantages over these alternatives.

1. *Behavior during demonstration will generally be high quality*, leaving only a potential period of tuning the learning algorithm as the source of low-quality interaction. In contrast, autonomous learning algorithms typically require long periods of low-quality behavior. This is especially true in complex domains where an accurate model of the effects of the robot's actions is unavailable. Robotic social interaction appears to be one such domain (though learning such a model is plausible). Low-quality behavior during social interaction is at best unhelpful and at worst disastrous. A robot with nonsensical behavior will probably fail to engage its human interaction partner, ending the potential for any future data for learning.
2. *Demonstrations show what the teacher would do when actually in the corresponding situation*. Direct programming, on the other hand, requires a person to imagine

what conditions should determine behavior and then designate what the robot should do in these various conditions. The following passage from Martin is a common sentiment within human subjects testing: "[surveys] can tell you only how people say they behave or what they say they think, not how they actually behave or what they actually think" (Martin 2008). Put differently, if you want to know what someone would do in a situation, you should put them in that situation and observe. Asking the person is unreliable. The same insight should apply for specifying robot behavior. A demonstrator can show what to do in a situation, whereas a traditional programmer can only imagine and tell. Autonomous learning presents different challenges to the human attempting to specify behavior. Specifying evaluation functions for multi-objective behavior is challenging, and social interaction is a particularly difficult multi-objective problem. For instance, a robot acting as an educational tutor could have a learning-related objective, but it also needs to balance its effects on the pupil's emotional state. Further, a long-term learning objective creates a large delay between behavior and informative outcomes, making autonomous learning even more difficult. Instead, some mix of short-term objectives would probably be required, including objectives related to the pupil's engagement, emotional state, short-term learning, behavior that supports long-term learning, and so on. Demonstration does not require this challenging specification of an evaluation function by an expert in autonomous learning algorithms.

3. *Demonstrations permit behavior specification by experts in the related domain (e.g., early childhood education), without needing an expert in robot software development or in autonomous learning algorithms to act as a middle man.* Therefore, an LfW framework permits wider deployment and more efficient behavior specification, assuming a predetermined learning algorithm that works effectively in a wide range of situations.

After describing related work in the following section, this paper presents and discusses the general algorithmic and experimental framework for learning from the wizard. In the final section before the conclusion, we provide two examples of research projects in progress that fit this framework: a social Turing test and a project to learn policies that enable a robot to act as a learning companion for a young child. The first project is intended to evaluate how well social behavior in general can be emulated. The second project is intended as a test of whether this paradigm can be applied to capture potential pedagogical benefit with human-controlled robot behavior, regardless of how indistinguishable the learned behavior is from the demonstrated behavior.

## Background on LfD

Learning from demonstration is a highly active area of research that is receiving growing attention from researchers. Here, we give a brief introduction to the formalisms of LfD and its common paradigms. We also discuss previous work that is relevant to our approach to using LfD for social interaction.

When a task's state is fully observable—i.e., adding more information to the immediate observation signal would not improve one's ability to predict the action—the result of LfD can be described as a behavioral policy, $\pi : S \times A \rightarrow [0, 1]$, that maps a state $s$ and an action $a$ to a probability of that action being chosen from the state. To learn the policy, the learning agent receives labeled data in the form of demonstrations from a teacher—often a human, though not necessarily. LfD is a special case of supervised learning. The demonstrations of the teacher provide labeled training instances, which can be used by different supervised learning algorithms to create a policy. In this fully observable setting, a demonstration $d_i$ provided by the teacher takes the form of a chronologically ordered sequence of state-action pairs, $d_i = \{(s_i^1, a_i^1), (s_i^2, a_i^2), ...\}$. To illustrate, 10 games of Tetris could be converted into 10 demonstrations, each with a sequence of Tetris board states and controller inputs at each of those states.

In many situations, however, the state is not fully observable. To perform LfD for tasks with state that is not observed, the learning target is a mapping function from the space of possible observation histories and the space of actions to a probability. One method for doing so is to manually design features drawn from the observation history (including past actions) and then to use those features as if they are fully observable state, performing learning from demonstration on produced feature-action pairs.

In their survey of robot learning from demonstration, Argall et al. (Argall et al. 2009) describe three common categories of algorithms for deriving a policy from demonstrations. Their three categories, generalized to fit learning social interaction from demonstration, are as follows. *Policy mapping* learns a direct mapping from the robot's observation history to action probabilities, using each demonstrated action and its corresponding observation history as a training sample for classification (for discrete actions) or regression (for continuous actions). *System model* approaches learn from demonstrations a transition function, a reward function, or both. When a reward function is derived from demonstrations, attempting to encapsulate the demonstrator's valuations of various behavior, the technique can be referred to as inverse reinforcement learning (Ng and Russell 2000; Abbeel and Ng 2004). Lastly, *planning* approaches learn elements of potential plans: pre- and post-conditions for actions. Under the planning framework, a policy consists of an action sequence and the pre- and post- conditions for undertaking each action. In many cases, state-action pair demonstrations may not provide enough information to learn a complete policy, thus demonstrations given in planning models typically include annotations, additional domain information that guides policy learning.

## Related work

We now give a review of previous work related to learning social behaviors. Schrum et al. (2011) describe an online software agent called UT^2 that used an evolutionary neural network to learn believable gameplay behaviors within a first-person shooter. This work reports the development and results of a software agent that learns to emulate human-style gameplay to win the BotPrize, an artificial intelligence competition in which judges attempt to identify humans and agents by observing their gameplay behavior. Of particular relevance to learning from the wizard was the UTˆ2 bot's use of human traces (recorded logs of human navigation behavior in the test environment) to reproduce certain patterns of behavior. Quantitative analysis revealed that agents that relied more heavily on human traces spent less time stuck in non-navigable situations and qualitative reports indicated that increased reliance on human traces led to judgements of smoother and more human-like behavior.

Maya Cakmak and Andrea Thomaz have developed algorithms that combined social behaviors into the LfD process. However, the social behaviors exhibited by the robot are mainly "social interfaces" for LfD. That is, the demonstrations are facilitated and enhanced by social behaviors, but the underlying behaviors learned by the algorithm were still manipulation tasks. (Cakmak and Thomaz 2012)

In Nikolaidis and Shah's cross-training framework (2013), a human and a robot perform a task together, switching roles in each iteration of training. From the demonstrations derived from the human's behavior in these sessions, the robot creates models of human behavior (directly in one role and via a reward function in another), with which the robot plans its future behavior. In this work the action space consists of task-oriented, manufacturing actions, such as placing screws or drilling a hole. The robot did not learn or exhibit intrinsically social behaviors, though the timing and action selection of the model may have faciliated social interaction as a byproduct. This work demonstrates that LfD can effectively develop non-social behavior for human-robot interaction.

Huang and Mutlu (2014) describe a method that uses dynamic Bayesian networks (DBNs) to learn a policy for social behavior generation. Human teachers gave demonstrations of a social narration task, which were labelled by hand to identify social speech, gaze, and gesture behaviors. The timings and order of these social behaviors were used as demonstrations to train the DBN. The learned DBN generated social behaviors for the robot, which in experiments increased ratings of naturalness, likeability, and effectiveness of robot social behavior in comparison to designer-specified behaviors and a DBN with the same structure but randomized parameters. The learned behavior here was not interactive and did not involve a Wizard of Oz setting, since the robot learned to deliver a scripted lecture, but this work nonetheless shares many aspects with learning from the wizard and provides evidence for its potential success.

To our knowledge, no previous work has used LfD to generate nonlinguistic, explicitly social actions during human-robot interaction.

## Algorithmic and experimental framework

In this section, we describe a general framework for learning social interaction from demonstration (i.e., Learning from the Wizard). We discuss both the algorithm and the experiment together, providing a unified description of three basic stages of conducting an experiment via LfW: Wizard-of-Oz demonstrations, development and application of the learning algorithm, and evaluation of the learned policy. Both the first and third of these stages require human participants.

With LfD, as is common with more traditional forms of supervised learning, the more closely the distribution of training data resembles that of the testing data, the better the learned policy will typically perform on test data. (For LfW, the test data is the robot performance when deployed autonomously.) Similarly, if the function that correctly maps input to output in the training environment differs from the one in the testing environment, it is particularly difficult to learn models that perform well on test data.

With learning from demonstration, if the demonstrator's presence is known to the interacting human participant, this awareness will undoubtedly affect the dynamics of the interaction. The test domain for robots in our paradigm is autonomous social interaction with a human. Thus having a demonstrator present during training and absent during testing might create a considerable difference in what robot behavior is most effective for interaction.

If so, participant awareness of the demonstrator would likely introduce a mismatch between the training and test environments and negatively impact the performance of the learning algorithm. Therefore, a Wizard of Oz approach to demonstrations is appropriate. Wizard of Oz (WoZ) refers to a scenario in which a robot is secretly controlled by a human puppeteer as it interacts with a human participant. In addition to creating WoZed demonstrations, we propose that the demonstrator's observations of the interaction should reflect the sensory input that the robot's learning algorithm and resultant behavioral policy will have available, subject to the constraint of making this input understandable to the demonstrator. This proposal follows a principle suggested by Crick et al. (Crick et al. 2011), who compared learning from demonstrations that were given by a person with either a full video stream or the color-segmented video stream that the robot used for localization. Human demonstrators with the color-segmented stream gave demonstrations of *lower* quality, but these demonstrations led to *better* learned performance. Though only one study has examined this principle, we find it intuitively appealing and that it matches other anecdotal accounts of LfD application.

Once sufficient demonstrations have been collected, a learning algorithm is applied to create a mapping from the space of possible observation histories and the space of possible actions to a probability. As mentioned in the Background section, many approaches to LfD break that mapping into two steps: mapping from observation history to features and mapping from features and an action to a probability. The second mapping might be addressed by standard policy mapping for LfD. The first mapping appears to be a particularly challenging problem in the context of learning social interaction.

The true state space of a person in social interaction—whether in person or controlling a robot—is unknown and intractably complex, since it involves the full neurological state that gives rise to their behavior. Social interaction involves memory and inference about information never observed (events, the other person's mood, etc.). . However, a mapping from the robot's observation history to a relatively simple state description might be sufficient to provide enough context to emulate the demonstrator's decisions with sufficient fidelity to produce the desired interaction effects (e.g., engagement, joy, and learning). The most common approach for deriving such a state signal from an observation history is to hand-design features that are thought to provide context for the demonstrator's action. Algorithmic approaches for deriving such state automatically may also be possible (e.g. deep-learning approaches (Sutskever and Hinton 2007)). .

When a satisfactory policy has been derived from demonstration data, the policy can be formally evaluated. We describe two categories of analysis, each of which is being explored by one of the two ongoing studies described in the following section. Evaluation metrics could include human participants' ratings of the robot attributes such as likability and naturalness; measurement of voluntary interaction time or number of interactions; or measurements of effects on the human interaction partner, including the person's mood, skill mastery, or subsequent behavior.

One type of evaluation directly compares the learned behavior to the demonstrations provided. Such a comparison provides an idea of how well the demonstration behavior was captured by LfD. The Social Turing Test study is an example of this type of evaluation.

A second type of evaluation compares the learned behavior to other types of interaction, which could include human-human interaction, interaction with a robot whose behavior was computationally derived from other techniques, and many other possibilities. The robot reading companion study employs this type of evaluation.

Above, we have assumed a batch approach to learning from only demonstrations. Alternatives exist and may be preferable. Many applications of LfD are iterative, allowing cycles of demonstration, testing the learned behavior, and further demonstration. Such an approach provides a simple way to decide when the number of demonstrations are sufficient for moving to the formal evaluation stage, and the learning curve throughout these iterations could itself be a form of evaluation. In iterative LfD, one common practice is to provide demonstrations from failure states. Initiating robot behavior from failure states is feasible in many tasks involving physical manipulation, but this approach seems infeasible for social interaction with a human, whose state cannot be specified by an experimenter. A potential substitute is to have a demonstrator provide control commands that are ignored during autonomous behavior, similar to pushing the buttons on a video-game remote as if playing while another person is actually in control. This method allows trajectories from novel states reached by the learned policy, states that might never be visited when the demonstrator is in control. Ross and Bagnell provide a theoretical treatment of this approach (Ross, Gordon, and Bagnell 2011). In addition to incremental alternatives to the three-stage description above, the robot could also learn from more than demonstrations. For instance, human-delivered feedback (Knox, Stone, and Breazeal 2013) could be given during autonomous behavior (Argall, Browning, and Veloso 2007).

## Current projects involving LfD for social HRI

To examine how LfD might be extended to learning social behaviors, we have developed a common platform to be used in two ongoing research projects, each intended to explore a different aspect of the learning process. However, we wish to make it clear that the details of the implementations below are only single instances of the general framework of LfW described above. The action and sensing spaces, interaction contexts, and evaluations described may be useful as guiding examples, but we believe that LfW as a paradigm is applicable to a much wider range of scenarios that involve learning social interaction behavior.

The first project attempts to learn a *believable* social behavior policy by engaging users in what we term a social Turing test. Turing's original test asked whether users could reliably determine whether text-based linguistic behavior was generated by a human or a computer during a short interaction. This *social* Turing test instead asks whether users can reliably determine whether an agent's *nonverbal* behavior is generated by a human or an autonomous agent during an interaction. Initially, we expect that humans will be able to easily deduce when the robot is being tele-operated. But as the system receives *a)* more teleoperated demonstrations and *b)* feedback on the autonomous policy (in the form of correctly identifying the robot as autonomous), we expect the robot's ability to emulate the teleoperator to improve significantly. The high-level research question in this project is whether it is possible to use logged behavioral data from teleoperated, human-controlled interactions to learn an autonomous behavior policy that can pass this social Turing test.

The second project attempts to learn pedagogically useful behavior policies to affect a child's language and literacy learning. In this project, there will be a more focused, structured interaction in which the robot and child will together explore a scene from a tablet-based picture e-book. Following the LfW framework, the robot's behavior will be tele-operated at first, then the logged interaction data will be used to learn an autonomous policy. The challenge here is to produce not only believable behavior that maintains a compelling interaction, but also behavior that can promote beneficial educational outcomes.

### Common robotic platform and cognitive architecture

In preparation for these projects, we have developed a robotic system composed of four major parts: the physical robot, a portable sensor suite, a teleoperator user interface, and an integrated cognitive architecture that coordinates sensing, learning, and shared control with a teleoperator.

For the physical robot, we used the Dragonbot. The Dragonbot is a socially expressive platform with 5 physical degrees of freedom (DOFs) and a face animated by the screen of an Android phone, capable of conveying many emotions

Figure 1: The Dragonbot, a socially expressive robot.

and expressions (Setapen 2012), as shown in Figure 1. We also have assembled a portable sensor package consisting of a Microsoft Kinect and a high-quality directional microphone to capture facial expressions and prosodic information from users. In order to facilitate demonstrations for the robot to learn from, we also developed a real-time teleoperator user interface for the system. The interface, shown in Figure 2 displays the robot's behavior by drawing the robot's joint configurations, shows the robot's sensory stream of audio and video features, provides controls for triggering robot actions, and allows the user to initiate and conclude "training sessions" for the robot. The robot can be controlled to take on persistent expressions, to execute short, emotional pre-scripted motor actions with paired audio, and to gaze at the human's face or other targets.

In order to realize a complete, integrated system capable of effectively making sense of large amounts of streaming data and real-time human user input, we also developed an integrated, cognitive architecture that receives streams of multi-modal data, logs features and actions, and handles seamless switching between teleoperation and autonomous control. The cognitive architecture is subject to change as these projects advance, but at the time of writing it operates as follows. Demonstrations are collected while the robot is controlled by a human wizard. At regular time intervals (currently 100 ms), the wizard's chosen action is executed—if an action has been chosen. To learn from these demonstrations, the robot creates a training sample for policy mapping from each step. This supervised-learning sample is derived by computing features from the observation history at that time step and considering the chosen action (or a *no-action* action if none was chosen) to be the label. Sample features include facial expression information and prosodic pitch and intensity information from the portable sensor suite, recently executed robot actions, and actions that the user has taken within the app (within the educational project). For supervised learning, we are focusing on classification algorithms that output a probability over classes. When the robot behaves autonomously, it computes these observation-based

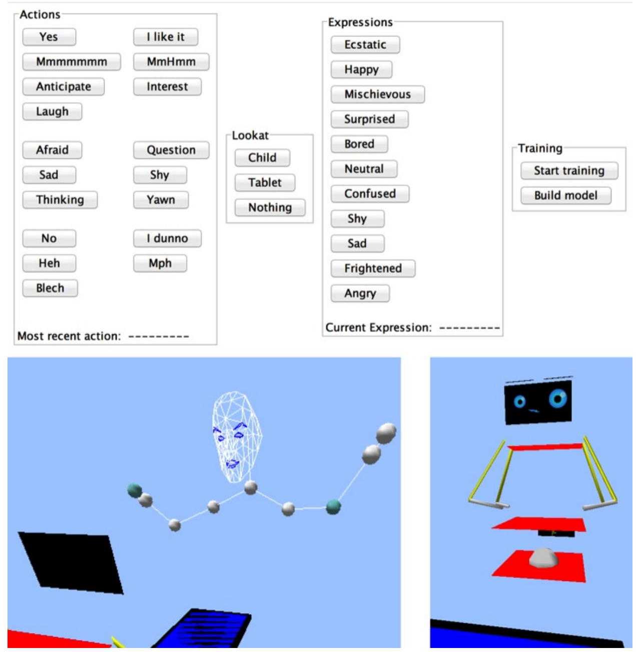


Figure 2: The GUI for Wizard-of-Oz demonstrations. This version, designed for the child-robot interaction in the robot learning companion project, contains a visualization of the tablet and an option to direct the robot's gaze at the tablet. The final version for this experiment will also include a visualization of app activity and buttons to trigger app-related robot actions.

features at each time step and samples an action from the probability distribution that the learned model outputs from these features. Our intuition is that non-deterministic behavior will be more believable, as it will display more variability, consistent with human-human interaction. A potential disadvantage, though, is that rare and unintended actions in demonstrations will likely occur. A deterministic policy would less likely to display these types of actions than a probabilistic one.

## A nonverbal, embodied, and social Turing Test

The social Turing test has some additional detail beyond what we have described above. First, human participants will have an interface with two buttons that lets them communicate whether they think the robot is controlled autonomously or by a human. Additionally, as mentioned before, human judgments of autonomous vs. human can be used as feedback on the interaction. To use this feedback, the learning algorithm above will need to be extended. In our evaluation of participants' judgment data, we expect to see a learning curve as the number of interactions increases, with the probability of an incorrect assessment increasing to some plateau. Reaching a probability near 0.5 would be a strong success, though lower plateaus could still provide evidence that the demonstrations and feedback improved the fidelity of the robot's autonomous imitation of human control.


Figure 3: The coloring scene from the interactive, illustrated e-book TinkRBook, which will be the focus of child-robot play during the educational interaction project.

### Sharing educational apps with children

In the robot learning companion study, participants will be interacting with the robot in the context of mutual play with an Android tablet app. The app we are using is part of an educational, interactive, illustrated e-book called TinkR-Book (Chang and Breazeal 2011) We will limit interaction to a scene in the e-book that involves mixing colors in a duck character's bath (Figure 3). The child can mix primary colors to make secondary colors and mix additional colors to make brown. The duck can also be washed clean to white to try other mixes. Additionally, the words are spoken when touched, and noun words highlight the corresponding object in the illustration. Likewise, touching the objects causes their noun word to be highlighted and spoken. During the interaction, the child's actions upon this app will be used as another source of context for robot behavior. Additionally, the robot's action set will be augmented to include gazing at the app, highlighting objects in the app, and triggering object touches in the app.

Child-robot-app interaction with the learned policy will be compared against the child interacting with the app alone. The primary evaluation metrics will be based on differences between pre-test and post-test responses to questions regarding the educational material in the e-book scene. In addition to these two conditions, the experiment may also include a child-adult-app interaction condition or even a child-adult-robot-app condition.

## Conclusion

In this paper, we have introduced a new model of LfD specifically focused on learning socially interactive behaviors, which we call learning from the wizard. In this scenario, the demonstrations are provided by a human tele-operating the robot in a Wizard-of-Oz paradigm. LfW provides several benefits for learning to interact socially: WoZ control makes it easier for the robot to avoid an early-learning period of low-quality behavior, demonstrations capture teacher behavior better than asking the teacher to describe behavior, and non-roboticist humans can easily provide demonstrations to the robot. This paper has presented a sketch of how to design an LfW system and perform experiments on it. Important questions remain that can only be addressed by actually conducting these experiments, including what features should be extracted from the observation history to provide behavioral context and how to choose action spaces that are rich enough for compelling interaction but simple enough to avoid requiring more demonstrations than are possible.

## References

Abbeel, P., and Ng, A. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, 1. ACM.

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

Argall, B.; Browning, B.; and Veloso, M. 2007. Learning by demonstration with critique from a human teacher. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, 57–64. ACM.

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

Chang, A., and Breazeal, C. 2011. Tinkrbook: shared reading interfaces for storytelling. In *Proceedings of the 10th International Conference on Interaction Design and Children*, 145–148. ACM.

Crick, C.; Osentoski, S.; Jay, G.; and Jenkins, O. C. 2011. Human and robot perception in large-scale learning from demonstration. In *Proceedings of the 6th international conference on Human-robot interaction*, 339–346. ACM.

Huang, C.-M., and Mutlu, B. 2014. Learning-based modeling of multimodal behaviors for humanlike robots. In *Proc. HRI*.

Knox, W. B.; Stone, P.; and Breazeal, C. 2013. Training a robot via human feedback: A case study. In *Social Robotics*. Springer. 460–470.

Martin, D. W. 2008. *Doing psychology experiments*. Cengage Learning.

Ng, A., and Russell, S. 2000. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc.

Nikolaidis, S., and Shah, J. 2013. Human-robot cross-training: Computational formulation, modeling and evaluation of a human team training strategy. In *Proceedings of the IEEE/ACM International Conference on Human-Robot Interaction*.

Ross, S.; Gordon, G. J.; and Bagnell, D. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*, 627–635.

Schrum, J.; Karpov, I. V.; and Miikkulainen, R. 2011. Ut 2: Human-like behavior via neuroevolution of combat behavior and replay of human traces. In *Computational Intelligence and Games (CIG), 2011 IEEE Conference on*, 329–336. IEEE.

Setapen, A. M. 2012. Creating robotic characters for long-term interaction.

Sutskever, I., and Hinton, G. E. 2007. Learning multilevel distributed representations for high-dimensional sequences. In *International Conference on Artificial Intelligence and Statistics*, 548–555.