Curriculum Design for Machine Learners in Sequential Decision Tasks

Bei Peng, James MacGlashan, Robert Loftin, Michael L. Littman, David L. Roberts, and Matthew E. Taylor

***Abstract*—Existing work in machine learning has shown that algorithms can benefit from the use of curricula—learning first on simple examples before moving to more difficult problems. This work studies the curriculum-design problem in the context of sequential decision tasks, analyzing how different curricula affect learning in a Sokoban-like domain, and presenting the results of a user study that explores whether non-experts generate effective curricula. Our results show that 1) the way in which evaluative feedback is given to the agent as it learns individual tasks does not affect the relative quality of different curricula, 2) non-expert users can successfully design curricula that result in better overall performance than having the agent learn from scratch, and 3) non-expert users can discover and follow salient principles when selecting tasks in a curriculum. We also demonstrate that our curriculum-learning algorithm can be improved by incorporating the principles people use when designing curricula. This work gives us insights into the development of new machine-learning algorithms and interfaces that can better accommodate machine- or human-created curricula.**

***Index Terms*—Curriculum Design; Curriculum Learning; Sequential Decision Tasks; Human-Agent Interaction**

I. Introduction

Humans acquire knowledge efficiently by starting from simple concepts, and then gradually generalizing to more complex ones using previously learned information. This learning strategy has been shown to be effective by a number of cognitive scientists, given that easier concepts can help shape the understanding of more complex ones [1]–[4]. Similar ideas are exploited in animal training [5]—animals can learn much better through progressive task shaping. Recent work [6]–[8] has shown that machine-learning algorithms can benefit from a similar training strategy, called *curriculum learning*. Rather than considering all training examples at once, the training data can be introduced in a meaningful order based on their apparent simplicity to the learner, such that the learner can build up a complex model step by step. The agent can learn faster on more difficult examples after it has mastered simpler examples. This approach was shown to drastically affect learning speed and generalization in supervised learning settings [6]–[8]. Major challenges in curriculum learning include determining how difficult a training example will be for the agent to learn and ensuring that each example presented to the agent is suitable given its current ability.

In most existing work, the curriculum is generated either automatically [7]–[10], by iteratively selecting examples with increasing difficulty tailored to the current ability of the learner, or manually by the algorithm designer, who will typically have specialized knowledge of the problem domain or of the algorithm itself [11]–[17]. How *non-expert humans* design curricula is currently a neglected topic.

We argue that this topic is a critical missing piece: a better understanding of the curriculum-design strategies used by non-expert humans may help us to 1) understand the general principles that make some curriculum-design strategies better than others, and 2) inspire the design of new machine-learning algorithms and interfaces that better accommodate the natural tendencies of human trainers. As more robots and virtual agents are deployed, more of their users will be non-experts, who will need to teach them new skills without programming them. This work focuses on understanding how non-expert human teachers design curricula and investigates how we can adapt machine-learning algorithms to better take advantage of this type of non-expert guidance. We believe this work is the first to explore how non-expert humans approach the design of curricula in sequential decision tasks and leverage its findings to improve a curriculum-aware machine-learning algorithm.

In this work, we study how the choice of curriculum affects an agent's ability to learn in our Sokoban-like test domain [18]. In this domain, each task is specified by a text command, and the agent is trained to perform the task via reward and punishment. Existing work has shown that hand coded [13] and agent-generated [9] curricula can speed up learning when the final (*target*) task is too difficult to be learned from scratch. In contrast, we study the effects of curricula when the agent can learn the target task, but may require more trainer interaction to do so. We also explore whether different approaches to teaching individual tasks affect the relative quality of curricula. Our results show that:

- Different curricula can have substantial impact on training speed (*i.e.*, the amount of data required to learn).
- Longer curricula can actually outperform shorter ones.
- Curriculum learning can be more beneficial as the target task's complexity increases.
- The relative performance of curricula is consistent across different methods for providing feedback to the agent.

To explore how non-experts generate curricula, we ran a human subjects experiment in which non-expert participants

B. Peng and M. E. Taylor are associated with the School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99164, USA bei.peng@wsu.edu, taylorm@eecs.wsu.edu

J. MacGlashan is associated with Cogitai jmacglashan@gmail.com

R. Loftin and D. L. Roberts are associated with the Department of Computer Science, North Carolina State University, Raleigh, NC 27695, USA rtloftin@ncsu.edu, robertsd@csc.ncsu.edu

M. L. Littman is associated with the Department of Computer Science, Brown University, Providence, RI 02912, USA mlittman@cs.brown.edu

This work has been reviewed by the WSU IRB, #13463-002

designed curricula for an agent. Analysis of these curricula shows that 1) participants can design curricula that result in better overall agent performance than learning from scratch, even when participants receive no feedback on the quality of their curricula, and 2) we can identify salient principles that participants follow when selecting tasks in a curriculum. We demonstrate that our curriculum-learning algorithm can be improved by incorporating some of these principles. We believe our results will be useful for the design of new machine-learning algorithms with inductive biases that favor the types of curricula non-expert humans typically provide.

II. Background and Related Work

Taylor et al. [13] first showed that curricula with different state descriptions and action spaces can be helpful in reinforcement learning (RL), transferring knowledge between increasingly complex tasks. In Bengio et al. [6], curricula were used for non-convex optimization problems in machine learning. That work points out that the notion of simple and complex tasks is often based on human intuition, and that there is value in understanding how humans identify "simple" tasks. Kumar et al. [7] and Lee et al. [8] each developed algorithms for supervised learning problems which automatically identify easy instances to learn from. Narvekar et al. [9] developed multiple methods to automatically generate novel tasks for curricula for multiagent RL domains, and Svetlik et al. [10] used potential-based shaping [19] to construct curricula from a set of source tasks. None of these works however look at the way in which humans actually design curricula.

We believe that non-expert users may be able to design good curricula by considering which examples are "too easy" or "too hard," given the learner's current understanding, similar to how humans are taught with the *zone of proximal development* [20]. Along these lines, Khan et al. [21] conducted studies in which human participants needed to teach a robot the concept of whether an object could be grasped with one hand. That work showed that human teachers can teach via a form of curriculum learning, specifically by starting with extreme instances that are far away from the decision boundary and then gradually approaching the boundary. In contrast, this work focuses on a somewhat different notion of curriculum learning, where the agent must understand multiple concepts to solve the target task, and a curriculum can be used to teach these concepts more efficiently. Our work also focuses specifically on sequential decision-making problems.

Finally, we note other paradigms in reinforcement learning are closely related, but not identical to, curriculum learning. Wilson et al. [22] explored multi-task learning in RL, where the agent needed to solve a number of Markov Decision Processes drawn from a common distribution. In multi-task learning however, the agent is evaluated on its performance across all tasks, with no specific training tasks. Transfer learning [23] resembles curriculum learning more closely in that knowledge from the source tasks is used to learn the target tasks more efficiently. Transfer learning methods generally assume that 1) the source tasks are predefined, 2) the agent knows nothing about the target tasks when learning the source tasks, and 3) the transfer of knowledge is a single-step process. Curriculum learning extends transfer learning to sequences of tasks presented in a specific order, that is, from simpler to more complex. Sutton et al. [24] explored the idea of lifelong learning [25] in the RL setting, considering the future sequence of tasks the agent could encounter. While both lifelong learning and curriculum learning involve a specific sequence of tasks, lifelong learning considers tasks that are not necessarily ordered so as to make learning more efficient. Active learning [26], in which the agent attempts to select the most informative examples for learning, has also been applied to RL domains [27], [28]. We can view the automatic construction of curricula as a form of active learning. Lastly, the notion of learning options [29] is related to our work in that it involves the learning of simpler skills that can be progressively combined into more complex behaviors.

III. Our Domain

To study whether non-expert humans can design good curricula, we used our Sokoban-like test domain. We choose this domain because it connects the learning of each task with a natural language model. Based on the language model, we can construct more complex tasks that depend on multiple simpler concepts that can be taught individually as part of a curriculum. For example, if the agent needs to learn the command "move the red bag to the yellow room," it must understand the concepts "red bag" and "yellow room." These concepts could be taught first using simpler tasks. The natural language description also allows humans to more easily isolate different concepts that the agent needs to learn to solve a task.

More specifically, our domain is a simple, simulated home environment of the kind shown in Fig. 1. The domain consists of four object classes: agent, room, object, and door. The agent is represented visually as a dog, since people are familiar with dogs being trained with feedback. The agent can move one unit in the four cardinal directions and push an object by moving into it. The objects are chairs, bags, backpacks, or baskets. Rooms and objects can be red, yellow, green, blue, or purple. Doors (shown in white in Fig. 1) allow the agent to move from one room to another. Therefore, the state space in this task includes the agent's location; rooms' locations and colors; objects' locations, colors, and shapes; and doors' locations.

Possible commands given to the agent include moving to a room (*e.g.*, "move to the red room") and taking a specified object to a room (*e.g.*, "move the red bag to the yellow room"). The agent learns to follow these text commands via a human or simulated trainer's reinforcement and punishment feedback. Our previous work [30], [31] found that non-expert humans are good at training the agent to execute new commands using reinforcement and punishment feedback. In this work, we focus on how humans designing curricula for an agent. Therefore, the reinforcement and punishment feedback will be given by a simulated trainer. As the simulated trainer teaches new tasks, the agent will (hopefully) become better at interpreting the language, thereby enabling the agent to successfully interpret and carry out novel commands without further training. For example, an agent might learn the interpretation of "red" and

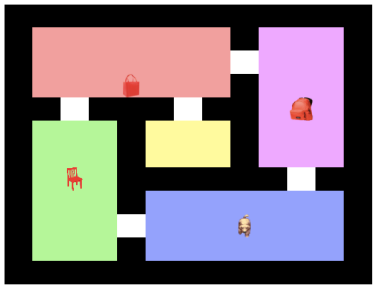


Fig. 1. The target environment #1 (command: "move the bag to the yellow room") used in our study has a dog, five colored rooms, and three objects.

"chair" from the command "move the red chair," and the interpretation of "blue" and "bag" from the command "bring me the blue bag," thereby allowing correct interpretation of the novel command "bring me the red bag." The simulated trainers are described further in Section V.

IV. Language Learning from Human Feedback

To enable language learning from agents trained with reward and punishment in our Sokoban-like test domain, we used a probabilistic model [18] that connected the IBM Model 2 (IBM2) language model [32] with a factored generative model of tasks, and the goal-directed Strategy-Aware Bayesian Learning (SABL) algorithm [18] for learning from feedback.

The SABL algorithm, which we developed in previous work [33], learns how different trainers use feedback, and then exploits that knowledge to accelerate learning. Instead of treating the feedback as a numerical reward signal [19], [34], SABL interprets it as a discrete communication that depends on both the trainer's desired behavior, and the training strategy they are using. SABL infers the desired behavior from the trainer's feedback using a probabilistic model of how humans provide feedback. This model assumes that a trainer will reward/reinforce (encourage), punish (discourage), or do nothing (neutral feedback), in response to each of the agent's actions. Under this model, reward is more likely when the agent takes a correct action, and punishment is more likely when it takes an incorrect one. SABL computes and follows a maximum likelihood estimate of the target policy given the feedback that the trainer has provided.

In the contextual bandit setting, SABL directly learns the policy for each state from human feedback. In large sequential domains, however, trainers may be more interested in communicating the final goals of a task. Learning these goals allows the agent to act correctly even in states where no feedback has been given. Assuming the agent has the ability to plan, the agent can simply take the action that is optimal for the known goal. We adapt SABL to the goal-directed setting [18] by representing goals as reward functions in an MDP, and computing the optimal policy π^g for any goal-based reward function $g \in G$. A "correct action" is then defined to be an action that is consistent with the optimal policy for the true goal: $a \in \pi^{g^\star(s)}$, where a is the action taken by the agent and $\pi^{g^\star}(s)$ is the set of optimal actions in state s for the true goal $g^\star \in G$. An "incorrect action" is an action that is inconsistent with the optimal policy: $a \notin \pi^{g^\star}(s)$.

V. Methodology

In our curriculum design problem, a sequence of n tasks, $M_1, M_2, \ldots, M_n$, must be selected. Each task M_i is defined by 1) an environment with an initial state s_i and 2) a text command e_i. The agent trains on these n tasks and then on the pre-defined target task, M_{n+1}. Here we define training speed as the number of trainer feedbacks required for the agent to learn to complete a task. A curriculum is successful if learning on task M_{n+1} is faster (fewer feedbacks required) with the curriculum than without it. A more difficult goal is to construct a sequence such that training on all $n+1$ tasks is faster than training directly on the final task, M_{n+1}.

We provide the curriculum designer with the 16 source tasks shown in Fig. 2.[1] For ease of description, we number the environments in the grid from 1 (top left) to 16 (bottom right) in English reading order. The 16 environments are organized along two dimensions: the number of rooms and the number of moveable objects. The cross product of these factors defines the overall complexity of the learning task, since these factors determine how many possible tasks the agent could execute in the environment and therefore how much feedback an agent could require to master its task. For example, Environment 1 has only a single possible task while in Environment 16 the agent may need to reach one of 5 rooms with 3 possible objects. Each environment includes a list of possible commands. For example, the possible commands in Environment 5 are "move to the red room," and "move the bag to the red room." Given the 16 room layouts, and the set of English language commands for each layout, there are 94 possible source tasks that could be included in a curriculum.

The target command is "move the bag to the yellow room." This command is not included for any source environment to disallow training directly on the target command. Furthermore, the target room layout is not one of the 16 layouts available. To study the effect of the target task's complexity on the performance of curricula, we use two target task layouts (Fig. 1 and Fig. 3), with the same command. We note that even though the number of possible tasks is the same in both environments, the second target task is harder than the first one because there are more competing hypotheses on the agent's way from the start state to the goal state in the second target task.

Our curriculum learning algorithm is shown in Algorithm 1. The algorithm takes the designed curriculum C, given target task M_{n+1}, and simulated trainer as input. The algorithm begins by initializing the IBM2 language model parameters arbitrarily and creating an initially empty training dataset D. Our algorithm learns each task in C in order, taking actions and receiving feedback for one task until the simulated trainer indicates that the agent is ready to move on to the next task.

For each task M_i, the agent is given the natural language command e_i, and must learn to perform the task defined by that command. Using the current IBM2 parameters, the prior distribution over all possible tasks τ is computed for the current environment and the command e_i. This task distribution

[1]These 16 environments largely cover the space of possible commands. Changing the layouts of these environments would not change how the language model interacts with them, and so should not significantly affect our results. The *construction* of source tasks is left to future work.

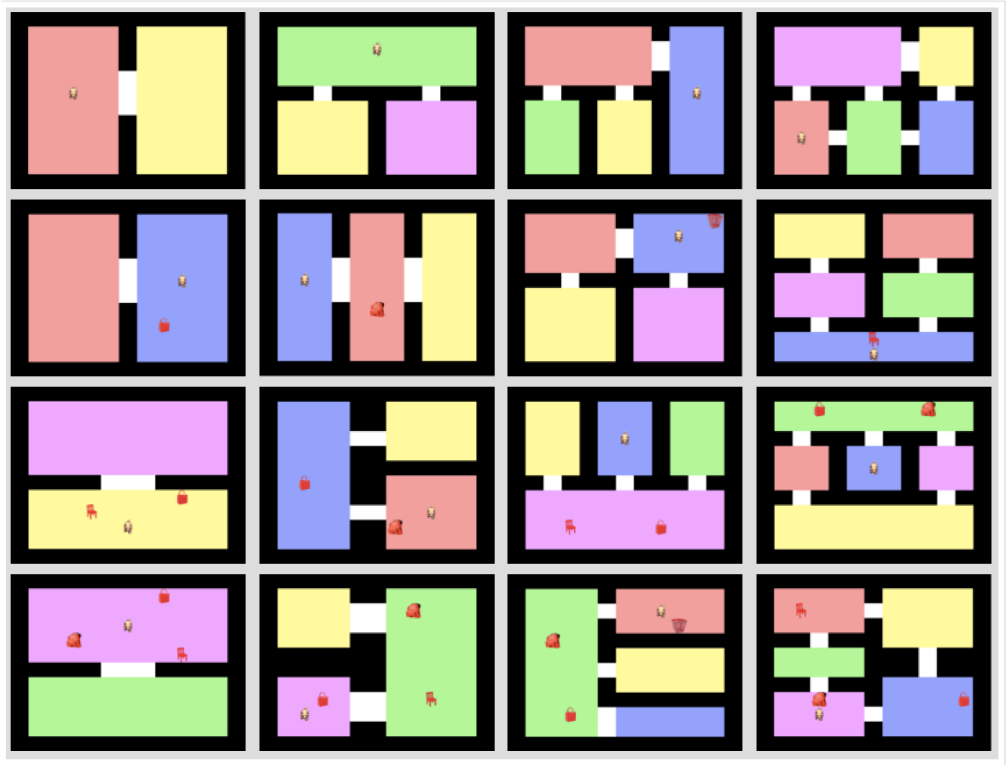


Fig. 2. The library of 16 environments is organized by the number of rooms and objects. There is a list of relevant commands for each environment.

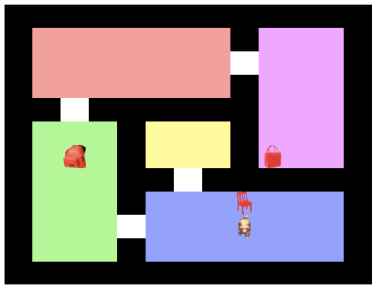


Fig. 3. The target environment #2 (command: "move the bag to the yellow room") used in our study has a dog, five colored rooms and three objects.

is used as a prior over goals in goal-directed SABL. To learn the correct goal, the agent first uses any "off-the-shelf" MDP planning algorithm to find the policy for each possible task. Next begins a behavior loop in which the agent follows the policy of the current most likely task, receives a feedback from the simulated trainer, and updates its belief in each task. This behavior loop continues until the simulated trainer terminates the session.[2] After completing training, for each possible machine language command m, a training instance is added to dataset D. Each training instance consists of the machine language command (m), the natural language command (e_i), and the posterior probability of m given the initial command e_i, and the state, action, and feedback sequence observed during the goal-directed SABL training process. Finally, the IBM2 parameters are retrained using the updated dataset D via weakly supervised learning.[3] After the language model is updated, training begins on the next task from the curriculum.

Algorithm 1 Curriculum Learning Algorithm

Input: curriculum $C = \{M_1, M_2, \ldots, M_n\}$, target task M_{n+1}, simulated trainer
Initialize IBM2 parameters arbitrarily
$D \leftarrow \{\}$ % Initialize training data
for i = 1 to $n+1$ **do**
  $(s_i, e_i) = (M_i.s_i, M_i.e_i)$
  $Pr(\tau) \leftarrow Pr(\tau | s_i, e_i)$
  repeat
    goalDirectedSABL(M_i)
  until simulated trainer terminates session
  for $m \in M$ **do**
    $D \leftarrow D \cup \{Pr(m|e_i, s, a, f), m, e_i\}$
  end for
  retrainIBM2Parameters(D)
end for

We consider three different simulated trainers, allowing us to study whether different methods for providing feedback to the agent influence which curricula are best. We focus on "explicit" feedback, where a trainer provides positive or negative feedback, as a proxy for trainer effort.

[2]Training for a task ends the first time the agent stops at the goal state.

[3]The training algorithm for the language model is an Expectation Maximization (EM) algorithm, as in prior work [35], similar to the standard EM algorithm for IBM Model 2, except that the contribution of each data instance is weighted by its posterior probability given the trainer feedback.

Correct trainer: Provide explicit, correct feedback for 50% of the agent's actions (*i.e.*, reinforcement for actions consistent with optimal policy, punishment otherwise).

Error-prone trainer: Provide explicit feedback for 50% of the agent's actions, with 20% of this feedback being incorrect[4], representing a worst-case scenario.

Entropy-driven trainer: Use the entropy of the agent's policy to target its feedback. This trainer provides correct feedback to 50% of actions where the entropy (H) of the agent's policy is high ($H > 0.1$) (*i.e.*, it is uncertain about the correct action in the current state). In state s, this entropy is:

$$H = -\sum_{a \in A} \Pr(a = a^\star | s, F) \ln(\Pr(a = a^\star | s, F)), \quad (1)$$

where A is the set of possible actions, F is the history of feedback from the trainer, and $\Pr(a = a^\star | s, F)$ is the probability, given F, that action a is an optimal action ($a^\star$) in state s. Actions with $H \leq 0.1$ never receive feedback.[5]

VI. Simulation Results

We first look at how different curricula affect the number of feedback signals the trainer must provide for the agent to learn just the target task, and to learn all the tasks in the curriculum. We hypothesized that 1) curricula could reduce the amount of feedback required to learn target task, 2) longer curricula would reduce the feedback required to learn more than shorter curricula, and 3) the reduction would be greater for more complex target tasks than for simpler ones.

As a baseline we created four sets of curricula of lengths $n = \{1, 2, 3, 4\}$. Each set contained 200 curricula generated by randomly selecting a sequence from the 16 environments in Fig. 2, and selecting a random command for each environment to define the learning task. Repetition of environments and tasks was allowed within a curriculum. We note that the number of possible curricula grows exponentially as the curriculum length increases. There are 94 possible curricula of length 1, $94 \times 94 = 8836$ possible curricula of length 2, and so on.

Each of these 800 curricula was evaluated 20 times for each of the three simulated trainers, and compared against directly learning the target task with that trainer. For both of the target tasks (shown in Figs. 1 and 3), we recorded the average amount of feedback required to learn 1) the target task, and 2) all tasks within the curricula (including the target task). Fig. 4 summarizes these results. As we expected, compared to directly learning the target tasks, all four sets of random curricula reduced the amount of feedback required to learn the target task itself (shown in Fig. 4(a) and Fig. 4(c)). Unpaired two sample t-tests [37] show that the differences in the amount of feedback required to learn the target task with and without the random curricula were statistically significant ($p \ll 0.01$), for each curriculum length and trainer combination. The absolute reduction was greater for the second, harder target task than for the first task, demonstrating that curricula can be more beneficial for more complex target tasks.

[4]Previous work [36] with real humans found an error rate of less than 10%.

[5]When $H \leq 0.1$, the probability of the most likely action is $> 99\%$ — additional feedback is unlikely to have much impact.

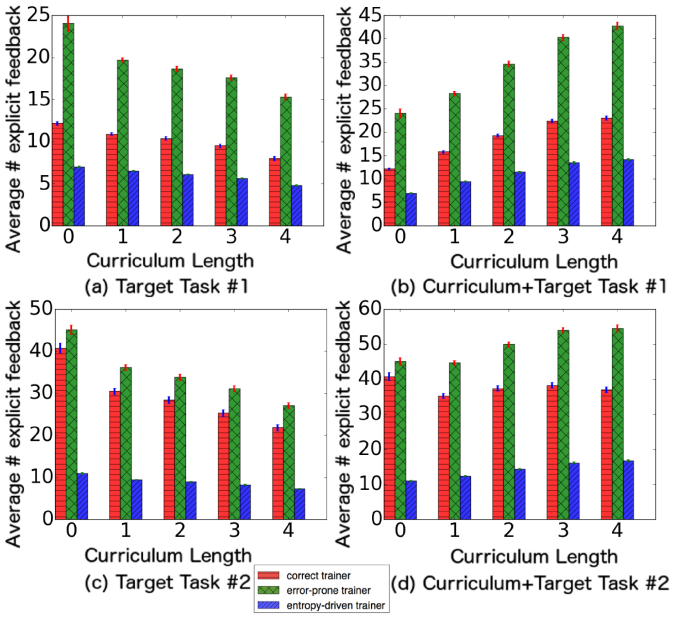


Fig. 4. Average number of explicit feedback signals needed to learn (a) target task #1, (b) all tasks (including target task #1), (c) target task #2, or (d) all tasks (including target task #2) on four sets of random curricula (or no curricula) with different simulated trainers. Error bars show standard errors.

We also found that longer curricula always reduced the amount of feedback required in the target task, relative to shorter curricula, in both target tasks. While we might expect longer curricula to require more feedback overall, we see in Fig. 4(d) that this is not always the case. Curricula of length 4 required fewer feedbacks than curricula of length 3, under the correct trainer. This shows that longer curricula can sometimes outperform shorter curricula in learning all tasks within the curricula (including the target task). As we expected (see Fig. 4), the type of simulated trainer used did not affect the relative quality of different curricula under these metrics.

Recall that the biggest challenge in curriculum design is to construct a sequence such that learning the entire curriculum (including the target task) is faster than directly learning the target task. As shown in Fig. 4(b), for the first task, none of the four sets of random curricula actually reduced on average the total amount of feedback required. However, Fig. 4(d) shows that for the harder target task, all four sets of random curricula did reduce the total amount of feedback required relative to directly learning the target task under the correct trainer. Unpaired two sample t-tests show that these differences were statistically significant ($p \ll 0.01$). This implies that as the target task's complexity increases, we may find more curricula which result in reduced total training time, while also reducing training time on the target task. However, even for the harder target task, we found that few curricula resulted in a lower total amount of feedback required under the error-prone trainer or entropy-driven trainer. We believe that the high probability of incorrect feedback with the error-prone trainer makes it difficult for the agent to fully leverage information from previous tasks in the curricula. For the entropy-driven trainer,

the amount of feedback required was already low relative to the other two trainers, making further improvement difficult.

VII. Human Subject Experiments

To study whether non-expert humans (Amazon Mechanical Turk workers in this case) can design good curricula, we conducted a series of experiments in which participants were asked to select a sequence of training tasks for the virtual dog to help it quickly learn to complete the final target task. Before the experiments, participants had to pass a color blindness test, after which they completed a background survey regarding their prior experience in training dogs. Participants then went through a tutorial that 1) walked them through two examples of the dog being trained to help them understand how the dog learns to perform a command from positive and negative feedback, and 2) taught them how to use the interface to design a curriculum. Participants were told "In this study, your goal is to design a curriculum (a set of assignments) for a virtual dog to train on, so that the dog can quickly complete the target assignment." They were also told that they could observe the process of the dog being trained on each task in their curriculum, including the target task, and that they would receive a bonus on top of their base compensation of $\$0.50$, depending on the quality of the curriculum they designed.

Following the tutorial, participants began the experiment itself, in which they selected environments and commands from a 16-environment grid (as in Fig. 2) in any order they wished to design a curriculum. Participants were required to include at least one source task in their curricula, but there was no upper limit on how long the curricula could be, and repeated tasks were allowed. The target task (Fig. 3) was always shown on the right side of the screen to remind the participant of what the agent ultimately needed to learn. We chose the more difficult target task for these studies since our previous results with random curricula suggested that curricula would have a larger effect on learning performance with this task, such that the effects of human-generated curricula might be more apparent. Once a participant finished their initial curriculum, they were shown the agent being trained (by the *correct* simulated trainer) on that curriculum, after which they were given the opportunity to redesign it to improve the agent's performance. Participants were required to redesign their curriculum at least once before making it their final submission, but could redesign it (and observe the training process) as many times as they wished before submitting.

To study the effects of the visual ordering of source tasks in the user interface, we conducted two experiments, each displaying the same 16 source tasks in a different layout:

- **Gradually Complex**: number of rooms increase from left to right, number of objects increase from top to bottom.
- **Gradually Simple**: number of rooms decrease from top to bottom, number of objects decrease from left to right.

The layout in the gradually simple condition was the transpose of that in the gradually complex condition, swapping Environments 1 and 16, 2 and 12, *etc.*, such that the difficulty of the environments gradually decreases from left to right, and top to bottom. Our first experiment used the gradually complex layout, while our second used the gradually simple layout.

We published these experiments on Amazon Mechanical Turk as a set of Human Intelligence Tasks. Between the two experiments we collected data from a total of 95 participants (95 unique AMT workers). Of these, we identified 15 participants whose completion time was less than 5 minutes (the average completion time was 22 minutes 18 seconds, with a standard deviation of 8.3 minutes) or who designed two curricula of length one. These 15 results were removed from the data, as the participants either did not understand the task, or were trying to maximize their payment per time unit, rather than attempting to design the curriculum well. Of the remaining data, there were 40 participants from each of the two experiments (gradually complex and gradually simple).

VIII. Results with Human Subjects

In evaluating the curricula designed by participants in our experiments, we consider both the initial and final curricula created by each participant (we ignore any redesigned curricula other than the final submission), and combine the curricula designed in both the first and second experiments into a single group, for a total of 160 human-generated curricula.[6]

A. Participant Performance

The goal in designing a curriculum is to allow the agent to learn the target task more quickly (with fewer trainer feedbacks) after going through the curriculum than it could just learning the target task directly. We therefore evaluate each human-generated curriculum by computing the average amount of feedback needed for the agent to learn the target task after being trained on the tasks in that curriculum, and the average amount of feedback required to learn the curriculum itself. Every curriculum was evaluated 20 times, with the agent having no knowledge from previous learning sessions.

Fig. 5(a) shows that, compared to directly learning target task #2, fewer feedbacks were required on average for the agent to master this target task after training on the human-generated curricula, under all three simulated trainers. Furthermore, Fig. 5(b) shows that fewer feedbacks in total were required for the agent to learn all tasks within the curricula (including target task #2) than learning the target task alone under the correct trainer. This demonstrates that human-generated curricula can actually achieve the goal of reducing the total effort required to teach the target task. A two-way ANOVA [37] shows that the differences in the amount of feedback required to learn the target task between using the curricula or not using the curricula were statistically significant ($p \ll 0.01$). The differences between the amount of feedback required under the three simulated trainers were also statistically significant ($p \ll 0.01$). Finally, interaction effects of these two factors on curriculum quality achieved were statistically significant ($p < 0.05$). Simple main effects analysis showed that significantly fewer feedbacks were required for the agent to master the target task after training on curricula than learning from scratch within each of the three trainer groups. It is worth noting that in our experiments participants

[6]The average curriculum length was 3.4, with a standard deviation of 2.0.

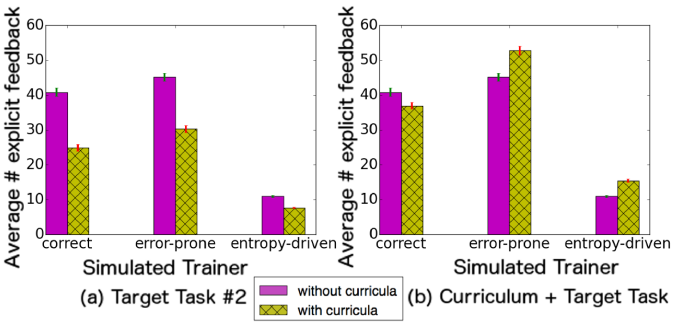


Fig. 5. Average number of explicit feedback signals needed to learn (a) target task #2 or (b) both the entire curriculum and target task #2, with and without human-generated curricula.

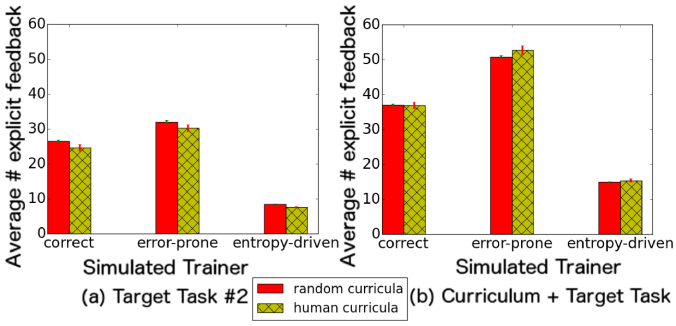


Fig. 6. Average number of explicit feedback signals needed to learn (a) target task #2 or (b) both the entire curriculum and target task #2, with random or human-generated curricula.

were not told exactly how much feedback was required for the agent to learn using their curricula, that is, the participants received no explicit feedback on the quality of their curricula.

In Section VI we saw that randomly generated curricula could lead to significant reductions in the amount of feedback required to learn the target task, and even reduce the total amount of feedback required to learn all the tasks in the curriculum. We therefore compared the human-generated curricula against randomly generated curricula, to see whether our non-expert participants could do better than simply selecting tasks at random. Specifically, we compared the average amount of feedback needed to learn the target task, and all the tasks (including the target task), after being trained either on a human-generated curriculum, or on one of the 800 randomly generated curricula used in experiment in Section VI.

These results (Fig. 6) show that human-generated curricula result in 1) fewer feedbacks required for the agent to master the target task, and 2) more feedback in total required for the agent to learn all tasks (including the target task), than random curricula. A two-way ANOVA shows that the differences in the amount of feedback required to learn the target task between using the human-generated curricula or the random curricula were statistically significant ($p < 0.01$). The feedback differences between the three simulated trainers were also statistically significant ($p \ll 0.01$). The interaction effects of these two factors on curriculum quality were not statistically significant ($p > 0.05$). This implies that non-expert humans did better than random in terms of improving the agent performance in learning the target task itself, but not in terms of reducing the overall effort required to teach the target task.

B. Curriculum Design Principles

One of the main goals of this work is to understand the general principles humans use when designing curricula, and to understand which design principles lead to the most effective curricula. We believe that such knowledge will inspire the development of new machine-learning algorithms which better accommodate the ways in which humans teach with curricula.

Recall that the command for the target task in our human-subjects experiments was "move the bag to the yellow room." This command itself was not available for any of the source environments to avoid curricula that simply had the agent learn the target command in a different environment. Commands did include the concepts "yellow room" and "bag" however, such that the agent could learn these required concepts before moving on to the target task. We found that in 72.5% of the human-designed curricula, more than half of the commands included at least one of the target concepts. In contrast, in only 55.8% of the random curricula were the target concepts present in more than half the commands. This suggests that participants preferred to teach the agent the target concepts, rather than other concepts like "blue room." In the background surveys, 67.3% of participants indicated that they had some dog training experience, which could partly explain this behavior.

Fig. 7 compares the performance of the human-generated curricula with more than half of the commands containing the target concepts against those with fewer such commands, under the correct trainer. We found that, for curricula with more target concepts, fewer feedbacks were required for the agent to learn 1) the final target task, and 2) all tasks (including the target task), relative to curricula with fewer of the target concepts. Unpaired two sample t-tests show that these differences were statistically significant ($p < 0.01$), demonstrating that the participants' strategy of selecting commands that include target concepts did in fact lead to better curricula.

C. Environment Preferences

We hypothesized that participants would prefer some source environments over others when designing their curricula. To understand how participants designed effective curricula, we examined these preferences by computing the fraction of participants who selected each of the 16 environments at least once, in either their initial or final curricula. Fig. 8 summarizes participants' preferences for each of the 16 environments when designing their initial and final curricula, with the ratios computed separately for the gradually complex and gradually simple grid layouts. The locations in these plots correspond to the positions of the environments they represent in the gradually complex grid (such that two dots in the same position in the two plots correspond to the same environment). A larger dot represents a higher probability of the corresponding environment being chosen. We found

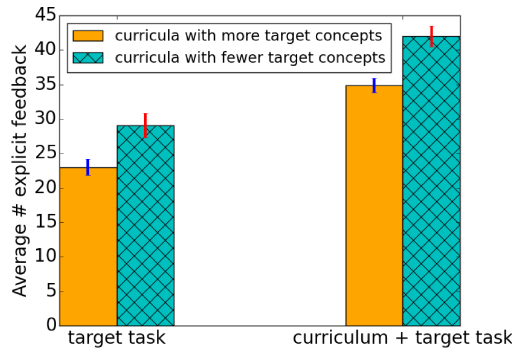


Fig. 7. Average number of explicit feedback signals needed to learn target task #2 or both the entire curriculum and target task #2 with human-generated curricula that contain more or fewer target concepts.

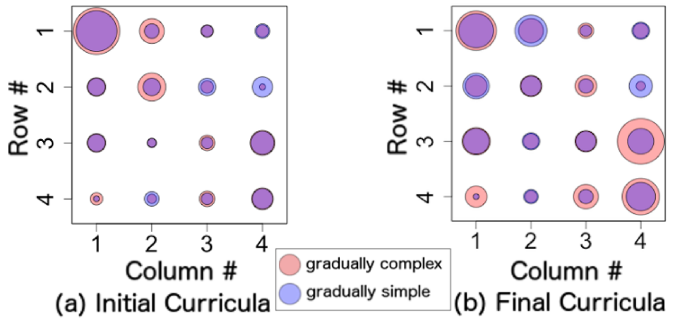


Fig. 8. The probability of each environment being included in a human-generated curriculum from both experimental conditions. The purple circle represents the overlap between conditions.

that when designing initial curricula, participants were more likely to select 1) Environments 1, 2, 6, 12, and 16 in the gradually complex condition, and 2) Environments 1, 12, and 16 in the gradually simple condition. This finding implies that participants preferred to choose the simplest environments that only contain one important concept (Environments 1 and 2 are the two simplest ones that refer to a yellow room, and Environment 6 is one of the two simplest ones that include an object) that the agent needed to learn for the target task, while also choosing more complex environments that more closely approximate the target environment. (Environments 12 and 16 are two of those most similar to the target environment.)

We also note that participants had a similar probability of choosing the two simplest environments (1 and 2) regardless of the layout of the user interface. Fisher's exact test shows that the differences in the frequencies of each of the 16 environments being selected by participants were not significantly different ($p > 0.05$) between the two experimental conditions, suggesting that the visual ordering of source environments does not influence participants' preferences. We believe that savvy participants prefer 1) isolating complexity, 2) selecting the simplest environments they can to introduce one complexity at a time, 3) choosing environments that are most similar to the target environment, and 4) introducing complexity by building on previous tasks rather than backtracking to introduce new types of complexity. These principles can be highly useful for the design of new machine-learning algorithms that better accommodate human teaching strategies.

Algorithm 2 Curriculum Learning from Non-expert Humans

Input: curriculum $C = \{M_1, M_2, \ldots, M_n\}$, target task M_{n+1}, simulated trainer, bias variable b, constant $B = 0.1$
Initialize IBM2 parameters arbitrarily
$D \leftarrow \{\}$ % Initialize training data
$K \leftarrow \{\}$ % Initialize learned concepts
for i = 1 to n **do**
  $(s_i, e_i) = (M_i.s_i, M_i.e_i)$
  $Pr(\tau) \leftarrow Pr(\tau | s_i, e_i)$
  repeat
    goalDirectedSABL(M_i)
  until simulated trainer terminates session
  for $m \in M$ **do**
    $D \leftarrow D \cup \{Pr(m|e_i, s, a, f), m, e_i\}$
  end for
  $K \leftarrow K \cup \{e_i.concepts\}$
  retrainIBM2Parameters(D)
end for
$k^* \leftarrow mostFreqConcepts(K)$
$(s_{n+1}, e_{n+1}) = (M_{n+1}.s_{n+1}, M_{n+1}.e_{n+1})$
$Pr(\tau) \leftarrow Pr(\tau | s_{n+1}, e_{n+1})$
$b = (1 + num(k^* \not\subseteq \tau.concepts) \times B)/num(\tau)$
for all $\tau' \in \tau$ **do**
  if $k^* \subseteq \tau'.concepts$ **then**
    $Pr(\tau') \leftarrow Pr(\tau') \times b$
  else
    $Pr(\tau') \leftarrow Pr(\tau') \times (b - B)$
  end if
end for
repeat
  goalDirectedSABL(M_{n+1})
until simulated trainer terminates session

D. Algorithm Improvement

A major goal of this work is to discover principles that humans use when designing curricula that can be leveraged to improve curriculum-learning algorithms. We saw in our experiments that most participants preferred to select task commands that include the target concepts, and that doing so results in curricula that lead to more efficient learning. Inspired by this, we investigated ways to bias the agent towards learning the concepts that the trainer taught the most in their curricula, which should have higher probability of being the target concepts the agent needs to learn in the final task.

In particular, our improved algorithm (Algorithm 2) summarizes the concepts K the agent learned in the entire curriculum C and gets the most frequently-learned concepts k^*. Then, when the agent moves to the target environment with initial state s_{n+1} and text command e_{n+1}, the prior probability for each task τ is multiplied by a bias variable b that is larger for tasks that include the most frequently-learned concepts k^* in the curriculum than for tasks that do not include them.

We implemented this improved algorithm and re-evaluated human-generated curricula under all three simulated trainers.

The result is shown in Fig. 9. We found that human-generated curricula could result in better agent performance in learning the target task using the improved algorithm versus using the original one, under all three simulated trainers. Similar results were found when considering the total amount of feedback required. A two-way ANOVA shows that the differences in the amount of feedback required to learn the target task between using the improved algorithm or using the original one were statistically significant ($p \ll 0.01$). The feedback differences between the three simulated trainers were also statistically significant ($p \ll 0.01$). The interaction effects of these two factors on curriculum quality achieved were statistically significant ($p < 0.05$). This suggests that the agent performance in learning the final target task can be significantly improved by biasing the final task towards the concepts the trainer taught the most in the curriculum.

We then compared the average amount of feedback needed for the agent to learn the target task, and all the tasks (including the target task), after being trained on human-generated curricula or on all four sets of random curricula using the improved algorithm (Fig. 10). We found that human-generated curricula resulted in fewer feedbacks required for the agent to learn the target task than random curricula, under all three simulated trainers. A two-way ANOVA shows that the differences in the amount of feedback required to learn the target task between using the human-generated curricula or using the random curricula were statistically significant ($p \ll 0.01$). The feedback differences between the three simulated trainers were also statistically significant ($p \ll 0.01$). The interaction effects of these two factors on curriculum quality achieved were statistically significant ($p \ll 0.01$). This demonstrates that human-generated curricula can be better than random for an agent using our improved algorithm.

IX. Conclusions and Future Work

In this paper, we investigated the curriculum design problem in the context of sequential decision tasks, where the goal was to design a sequence of source tasks for an agent to train on such that the agent can complete a pre-specified target task quickly with minimal explicit feedback. We analyzed how different curricula influence agent learning in a Sokoban-like household domain. Our results show that 1) different curricula can have substantial impact on training speed, while longer curricula can sometimes outperform shorter curricula in learning all tasks within the curricula (including the target task), 2) more benefits of curricula can be found as the target task's complexity increases, and 3) the way in which evaluative feedback is given to the agent as it learns individual tasks does not affect the relative quality of different curricula. We also present an empirical study designed to explore how non-expert humans generate such curricula. We show that 1) participants can successfully design curricula that result in better overall agent performance than learning from scratch, even when participants receive no feedback on the quality of their curricula, and 2) we can identify salient principles that participants follow when selecting tasks in a curriculum. We demonstrate that our curriculum-learning algorithm can be improved by incorporating some of these principles.

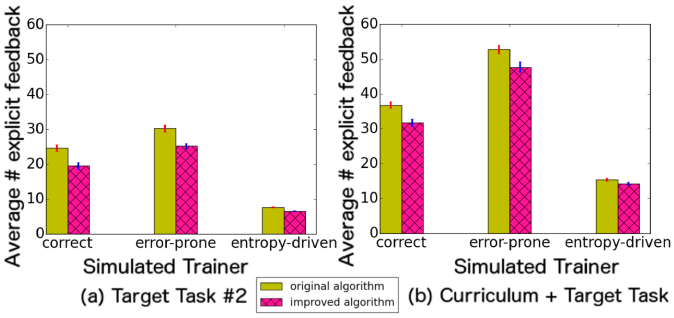


Fig. 9. Average number of explicit feedback signals needed to learn (a) target task #2 or (b) both the entire curriculum and target task #2 with human-generated curricula using the improved algorithm vs. using the original one.

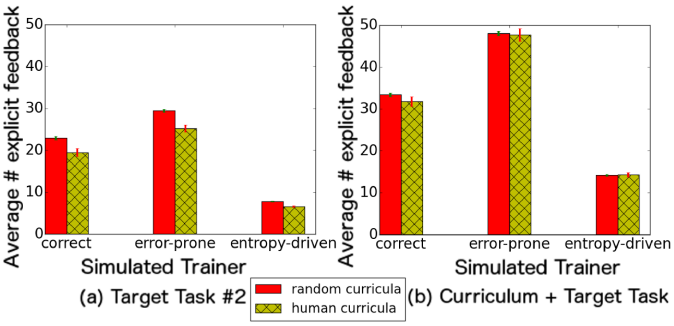


Fig. 10. Average number of explicit feedback signals needed to learn (a) target task #2 or (b) both the entire curriculum and target task #2 with random and human-generated curricula using the improved algorithm.

Considering that real world tasks are likely to be harder, we can speculate on ways of generalizing our findings to more complex task domains. First, given the finding that the reward feedback strategy does not change which curricula are best, we could choose the feedback strategy that minimizes the number of actions needed for the agent to complete the more complex task (*e.g.*, robot navigation tasks), where faster training time is of critical importance. Second, we could incorporate the salient principles (*e.g.*, isolating complexity) we found about humans when designing curricula into the automatic process of generating useful source tasks in any task domain. Third, the interface design could be improved to guide the non-experts to design better curricula.

Future work will study curriculum design where 1) participants can *create* a sequence of novel source tasks for the agent to train on, and 2) participants can see a score of the designed curricula and use this feedback in their design process, and 3) the learning algorithm is able to leverage more patterns used by non-expert curricula designers.

Acknowledgements

This research has taken place in part at the Intelligent Robot Learning (IRL) Lab, Washington State University and the CIIGAR Lab at North Carolina State University. IRL research is supported in part by NSF IIS-1149917. CIIGAR research is supported in part by NSF IIS-1643411.

REFERENCES

[1] B. Skinner, "The behavior of organisms: an experimental analysis." 1938.

[2] J. L. Elman, "Learning and development in neural networks: The importance of starting small," *Cognition*, vol. 48, no. 1, pp. 71–99, 1993.

[3] G. B. Peterson, "A day of great illumination: Bf skinner's discovery of shaping," *Journal of the Experimental Analysis of Behavior*, vol. 82, no. 3, pp. 317–328, 2004.

[4] K. A. Krueger and P. Dayan, "Flexible shaping: How learning in small steps helps," *Cognition*, vol. 110, no. 3, pp. 380–394, 2009.

[5] B. F. Skinner, "Reinforcement today." *American Psychologist*, vol. 13, no. 3, p. 94, 1958.

[6] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.

[7] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Advances in Neural Information Processing Systems*, 2010, pp. 1189–1197.

[8] Y. J. Lee and K. Grauman, "Learning the easy things first: Self-paced visual category discovery," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1721–1728.

[9] S. Narvekar, J. Sinapov, M. Leonetti, and P. Stone, "Source task creation for curriculum learning," in *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems*, Singapore, May 2016.

[10] M. Svetlik, M. Leonetti, J. Sinapov, R. Shah, N. Walker, and P. Stone, "Automatic curriculum graph generation for reinforcement learning agents," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2016.

[11] B.-H. Yang and H. Asada, "Progressive learning and its application to robot impedance learning," *IEEE transactions on neural networks*, vol. 7, no. 4, pp. 941–952, 1996.

[12] K. O. Stanley, B. D. Bryant, and R. Miikkulainen, "Evolving neural network agents in the nero video game," in *Proceedings of the IEEE 2005 Symposium on Computational Intelligence and Games (CIG'05)*. Piscataway, NJ: IEEE, 2005.

[13] M. E. Taylor, P. Stone, and Y. Liu, "Transfer Learning via Inter-Task Mappings for Temporal Difference Learning," *Journal of Machine Learning Research*, vol. 8, no. 1, pp. 2125–2167, 2007.

[14] A. Karpathy and M. Van De Panne, "Curriculum learning for motor skills," *Advances in Artificial Intelligence*, pp. 325–330, 2012.

[15] D. Abel, J. Salvatier, A. Stuhlmüller, and O. Evans, "Agent-agnostic human-in-the-loop reinforcement learning," *arXiv preprint arXiv:1701.04079*, 2017.

[16] A. Clegg, W. Yu, Z. Erickson, C. K. Liu, and G. Turk, "Learning to navigate cloth using haptics," *arXiv preprint arXiv:1703.06905*, 2017.

[17] C. Tessler, S. Givony, T. Zahavy, D. J. Mankowitz, and S. Mannor, "A deep hierarchical approach to lifelong learning in minecraft." in *AAAI*, 2017, pp. 1553–1561.

[18] J. MacGlashan, M. L. Littman, R. Loftin, B. Peng, D. L. Roberts, and M. E. Taylor, "Training an agent to ground commands with reward and punishment," in *Proceedings of the AAAI Machine Learning for Interactive Systems Workshop*, 2014.

[19] A. Y. Ng, D. Harada, and S. J. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Proceedings of the Sixteenth International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., 1999, pp. 278–287.

[20] L. S. Vygotsky, *Mind in Society: Development of Higher Psychological Processes*. Harvard University Press, 1978.

[21] F. Khan, B. Mutlu, and X. Zhu, "How do humans teach: On curriculum learning and teaching dimension," in *Advances in Neural Information Processing Systems*, 2011, pp. 1449–1457.

[22] A. Wilson, A. Fern, S. Ray, and P. Tadepalli, "Multi-task reinforcement learning: a hierarchical bayesian approach," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 1015–1022.

[23] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *The Journal of Machine Learning Research*, vol. 10, pp. 1633–1685, 2009.

[24] R. S. Sutton, A. Koop, and D. Silver, "On the role of tracking in stationary environments," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 871–878.

[25] S. Thrun, "Is learning the n-th thing any easier than learning the first," in *Advances in Neural Information Processing Systems*, vol. 8, 1996, pp. 640–646.

[26] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *Journal of artificial intelligence research*, 1996.

[27] L. Mihalkova and R. J. Mooney, "Using active relocation to aid reinforcement learning." in *FLAIRS Conference*, 2006, pp. 580–585.

[28] A. Baranes and P.-Y. Oudeyer, "Active learning of inverse models with intrinsically motivated goal exploration in robots," *Robotics and Autonomous Systems*, vol. 61, no. 1, pp. 49–73, 2013.

[29] C. M. Vigorito and A. G. Barto, "Intrinsically motivated hierarchical skill learning in structured environments," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 2, pp. 132–143, 2010.

[30] B. Peng, R. Loftin, J. MacGlashan, M. L. Littman, M. E. Taylor, and D. L. Roberts, "Language and policy learning from human-delivered feedback," in *Proceedings of the Machine Learning for Social Robotics workshop (ICRA)*, 2015.

[31] B. Peng, J. MacGlashan, R. Loftin, M. L. Littman, D. L. Roberts, and M. E. Taylor, "A need for speed: Adapting agent action speed to improve task learning from non-expert humans," in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2016, pp. 957–965.

[32] P. F. Brown, J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin, "A statistical approach to machine translation," *Computational linguistics*, vol. 16, no. 2, pp. 79–85, 1990.

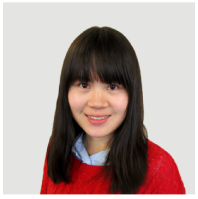
[33] R. Loftin, B. Peng, J. MacGlashan, M. L. Littman, M. E. Taylor, J. Huang, and D. L. Roberts, "Learning behaviors via human-delivered discrete feedback: modeling implicit feedback strategies to speed up learning," *Autonomous Agents and Multi-Agent Systems*, vol. 30, no. 1, pp. 30–59, 2016.

[34] W. B. Knox and P. Stone, "Reinforcement learning from simultaneous human and MDP reward," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 2012, pp. 475–482.

[35] J. MacGlashan, M. Babes-Vroman, M. DesJardins, M. Littman, S. Muresan, and S. Squire, "Translating english to reward functions," Technical Report CS14-01, Computer Science Department, Brown University, Tech. Rep., 2014.

[36] R. Loftin, B. Peng, J. MacGlashan, M. L. Littman, M. E. Taylor, J. Huang, and D. L. Roberts, "Learning something from nothing: Leveraging implicit human feedback strategies," in *Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on*. IEEE, 2014, pp. 607–612.

[37] T. H. Wonnacott and R. J. Wonnacott, *Introductory statistics*. Wiley New York, 1972, vol. 19690.



Bei Peng is currently a PhD student at the IRL lab of Washington State University. Her research mainly focuses on interactive machine learning, reinforcement learning, and curriculum learning. She is interested in studying how non-expert human teachers want to teach the agent to learn new complex tasks and how to incorporate these insights into the development of new machine learning algorithms. She has also worked on curriculum learning, while focusing on studying how non-expert humans approach designing curricula for the agent.

James MacGlashan is currently a research scientist at Cogitai, Inc. Before that, he was a postdoctoral researcher at Brown University in computer science after receiving his PhD in computer science from the University of Maryland, Baltimore County in 2013. His research spans a large range of artificial intelligence topics, though he primarily is involved in reinforcement learning and autonomous planning research. He does research in methods for artificial agents to learn from human teachers in various ways and he is the creator of the Brown-UMBC Reinforcement Learning and Planning (BURLAP) Java library.

Robert Loftin is currently a PhD student at the CIIGAR lab of North Carolina State University. His research focuses on developing algorithms for reinforcement learning in complex domains. He has worked on algorithms for transfer learning in RL domains, as well as the application of RL domains with continuous and high dimensional state spaces. His recent work has looked at developing RL algorithms that are stable when planning over long time horizons. He has also done work on the problem of computers learning from human teachers. Specifically, he has looked at ways that computers can learn more effectively from positive and negative feedback.

Michael L. Littman , Professor of Computer Science at Brown University, carries out research in machine learning and decision making under uncertainty. He is co-director of Brown's Humanity Centered Robotics Initiative and a Fellow of the Association for the Advancement of Artificial Intelligence.

David L. Roberts is an Associate Professor in the Computer Science Department at NC State. His current research interests are generally in the area of machine learning and artificial intelligence and their applications to the design of interactive technological experiences such as computer games, interactive dramas, or training scenarios. He focuses on these experiences as playable artifacts and therefore is interested in the in situ evaluation of his research. He incorporates concepts from social and behavioral psychology into his research as well. The designs of his algorithms are informed by concepts from psychology that can enable efficient and intuitive authorial and gameplay experiences.

Matthew E. Taylor received his doctorate from the Department of Computer Sciences at UT-Austin in 2008. Matt then completed a two-year postdoc at the University of Southern California and was an assistant professor at Lafayette College. He holds the Allred Distinguished Professorship in Artificial Intelligence at Washington State University in the School of EECS and is a recipient of the National Science Foundation CAREER award. Matt is currently on leave at Borealis AI, a Canadian institute funded by the Royal Bank of Canada, where he leads a research team focused on reinforcement learning.