Curriculum Design for Machine Learners in Sequential Decision Tasks

Bei Peng
Washington State University
beipeng.peng@gmail.com

James MacGlashan
Brown University
jmacglashan@gmail.com

Robert Loftin
North Carolina State University
rtloftin@ncsu.edu

Michael L. Littman
Brown University
mlittman@cs.brown.edu

David L. Roberts
North Carolina State University
robertsd@csc.ncsu.edu

Matthew E. Taylor
Washington State University
taylorm@eecs.wsu.edu

ABSTRACT

Existing machine-learning work has shown that algorithms can benefit from curricula—learning first on simple examples before moving to more difficult examples. This work defines the curriculum-design problem in the context of sequential decision tasks, analyzes how different curricula affect agent learning in a Sokoban-like domain, and presents results of a user study that explores whether non-experts generate such curricula. Our results show that 1) different curricula can have substantial impact on training speeds while longer curricula do not always result in worse agent performance 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, 3) the method for providing reward feedback to the agent as it learns within a curriculum does not change which curricula are best, 4) non-expert users can successfully design curricula that result in better overall agent performance than learning from scratch, even in the absence of feedback, and 5) non-expert users can discover and follow salient principles when selecting tasks in a curriculum. This work gives us insights into the development of new machine-learning algorithms and interfaces that can better accommodate machine- or human-created curricula.

Keywords

Curriculum Design; Curriculum Learning; Sequential Decision Tasks; Human-Agent Interaction; Crowdsourcing Experiments

1. INTRODUCTION

Humans acquire knowledge efficiently through a highly organized education system, starting from simple concepts, and then gradually generalizing to more complex ones using previously learned information. Similar ideas are exploited in animal training [16]—animals can learn much better through progressive task shaping. Recent work [2, 9, 10] 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 more complex model step by step. The agent will be able to learn faster on more difficult examples after it has mastered simpler examples. This training strategy was shown to drastically affect learning speed and generalization in supervised learning settings.

While most existing work on curriculum learning (in the context of machine learning) focuses on developing automatic methods to iteratively select training examples with increasing difficulty tailored to the current ability of the learner, how *humans* design curricula is one neglected topic. A better understanding of the curriculum-design strategies used by humans may help us design machine-learning algorithms and interfaces that better accommodate natural tendencies of human trainers. Another motivation for this work is the increasing need for non-expert humans to teach autonomous agents new skills without programming. Published work in Interactive Reinforcement Learning [5, 7, 8, 17, 18, 24] has shown that reinforcement learning (RL) [19] agents can successfully speed up learning using human feedback, demonstrating the significant role humans play in teaching an agent to learn a (near-) optimal policy. As more robots and virtual agents become deployed, the majority of teachers will be non-experts. This work focuses on understanding non-expert human teachers rather than finding the most efficient way to solve our sequential decision problem—future work will investigate how to 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 designing curricula in the context of sequential decision tasks.

In this work, we introduce and define the curriculum design problem in the context of sequential decision tasks. In our sequential decision domain, an agent must learn tasks in a simulated home environment. The tasks are specified via text commands and the agent is trained with reinforcement and punishment. The goal of a curriculum is to allow an agent to improve learning.

We are interested in studying how different curricula affect agent learning in our Sokoban-like test domain [12]. Existing work [14] has shown that a multistage curriculum can speed up learning when the final (*target*) task is too difficult for the agent to learn from scratch, we aim to explore the effect of curricula when the target task is not too hard to directly learn. We hypothesize that more benefits of curricula could

be found as the complexity of the target task increases. We also explore whether the best curricula change as agents are trained differently. Our results show that:

- Different curricula can have substantial impact on training speeds while longer curricula do not always result in worse agent performance in learning all tasks within the curricula (including the target task).
- More benefits of curricula can be found as the target task's complexity increases.
- The method for providing reward feedback to the agent as it learns within a curriculum does not change which curricula are best.

To explore how non-experts generate curricula, we task non-expert humans with designing a curriculum for an agent and evaluate the curricula they produce. The user-study results show that non-expert users can 1) successfully design curricula that result in better overall agent performance than learning from scratch, even in the absence of feedback on their quality, and 2) discover and follow salient principles when selecting tasks in a curriculum. We believe these results will be useful for the design of new machine-learning algorithms with inductive biases that favor the types of curricula non-expert human teachers use more frequently.

2. BACKGROUND AND RELATED WORK

The concept of curriculum learning was proposed by Bengio et al. [2] to solve the non-convex optimization task in machine learning more efficiently. They pointed out that the way we define curriculum strategies leaves a lot to be defined by human teachers. Motivated by their work, considering the case where it is hard to measure the easiness of examples, Kumar et al. [9] developed a self-paced learning algorithm to select a set of easy examples in each iteration, to learn the parameters of latent variable models in machine learning tasks. Lee et al. [10] proposed a self-paced approach to solve the visual category discovery problem by self-selecting easier instances to discover first, gradually discovering increasingly complex models. To study the teaching strategies followed by humans, Khan et al. [6] conducted behavioral studies where humans need to teach a target concept with a simple 1D threshold to a robot, and showed that human teachers follow the curriculum learning principle—starting with extreme instances that are farther away from the decision boundary and then gradually approaching it.

Although previous work has shown that machine-learning algorithms can benefit from curriculum strategies, there is limited work on curriculum learning in the context of sequential decision tasks. Wilson et al. [28] explored the problem of multi-task RL, where the agent needed to solve a number of Markov Decision Processes drawn from the same distribution to find the optimal policy. Sutton et al. [20] extended the idea of lifelong learning [25] to the RL setting, considering the future sequence of tasks the agent could encounter. Both cases assume a sequence of RL tasks is presented to a learner and the goal is to optimize over all tasks rather than only the target task. The idea of active learning [4] was exploited in RL domains [1, 13] to actively maximize the rate at which an agent learns its environment's dynamics. Options learning in hierarchical reinforcement learning [26] has also been shown to be able to enable the agent to develop progressively more complex skills.

Of existing RL paradigms, transfer learning [22] is the most similar to curriculum learning. The main insight behind transfer learning is that knowledge learned in one or more source tasks can be used to improve learning in one or more related target tasks. However, in most transfer learning methods: 1) the set of source tasks is assumed to be provided, 2) the agent knows nothing about the target tasks when learning source tasks, and 3) the transfer of knowledge is a single-step process and can be applied in similar domains. In contrast, curriculum learning aims to use a sequence of tasks so that an agent can develop progressively more complex skills and improve performance on a pre-specified target task.

Taylor et al. [23] first showed that curricula work in RL via transfer learning by gradually increasing the complexity of tasks. Narvekar et al. [14] developed a number of different methods to automatically generate novel source tasks for a curriculum, and showed that such curricula could be successfully used for transfer learning in multiagent RL domains. Svetlik et al. [21] proposed to use reward shaping [15] to automatically construct effective curricula given a set of source tasks. However, none of their work investigates human-designed curricula. We believe non-expert users may be able to design successful curricula by considering which examples are "too easy" or "too hard," similar to how humans are taught with the *zone of proximal development* [27].

3. LANGUAGE LEARNING FROM HUMAN FEEDBACK

To enable an artificial agent to effectively carry out a variety of different tasks with reward and punishment, an interface should connect the task learning with a language model. In our setting, a simulated trainer could give a novel command and reward and punish the agent until the agent successfully completed the task. As the simulated trainer taught additional tasks, the agent would be better at interpreting the language, thereby enabling the agent to successfully interpret and carry out novel commands without any reward and punishment. For example, an agent might learn the interpretation of "red" and "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."

To enable language learning from agents trained with reward and punishment, we used a probabilistic model [12] that connected the IBM Model 2 (IBM2) language model [3] with a factored generative model of tasks, and the goal-directed SABL algorithm [11] for learning from feedback. In SABL, trainer feedback is modeled as random variables that depend on the trainer's desired policy and the agent's last action. The trainer feedback model assumes that a trainer will reward, punish, or do nothing (neutral feedback), in response to the agent taking a correct or incorrect action, with respect to the task they are training. Given an MDP, an action is assumed to be correct if it is an optimal action for the MDP in the current state, and incorrect otherwise. In general, reinforcements under this model are more likely when the agent selects a correct action, and punishments are more likely when the action was incorrect. Using this model of feedback, SABL computes and follows the maximum likelihood estimate of the trainer's target policy given

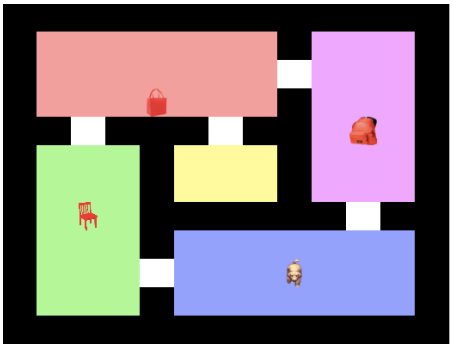

Figure 1: The target environment #1 (command: "move the bag to the yellow room") used in our study has a dog, five colored rooms and three objects (chair, bag, and backpack).

the history of actions taken and the feedback that the trainer has provided. We adapted SABL to this goal-directed setting by assuming that goals are represented by MDP reward functions and that the agent uses a planning algorithm to compute optimal policies for goal-based reward functions. Then, SABL's typical formulation of a "correct action" is redefined to be an action that is consistent with the optimal policy of the true goal being trained.

4. METHODOLOGY

In this section, we first describe our sequential decision domain for command learning. Second, we define the curriculum design problem and our application of it. Third, we discuss learning tasks in this domain.

4.1 Our Domain

Our domain is a simplified simulated home environment of the kind shown in Figure 1. The domain consists of four object classes: agent, room, object, and door. The visual representation of the agent is a virtual dog, since people are familiar with dogs being trained with reinforcement and punishment. The agent can deterministically move one unit in the four cardinal directions and pushes objects by moving into them. The objects are chairs, bags, backpacks, or baskets. Rooms and objects can be red, yellow, green, blue, and purple. Doors (shown in white in Figure 1) connect two rooms so that the agent can move from one room to another. Therefore, the state space in this task includes the agent's location; rooms' location and color; objects' location, color and shape; and doors' location. The 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 an automated trainer's reinforcement and punishment feedback.

4.2 Curriculum Design

In curriculum learning, the goal is to generate a sequence of n tasks, $M_1, M_2, \ldots, M_n$, for an agent to train on. The agent should train on these n tasks and then train on the pre-defined target task, M_t. The curriculum is successful if learning on task M_t is faster with the curriculum than without it. A more difficult goal is to construct a sequence such that training on the entire $n+1$ tasks is faster than

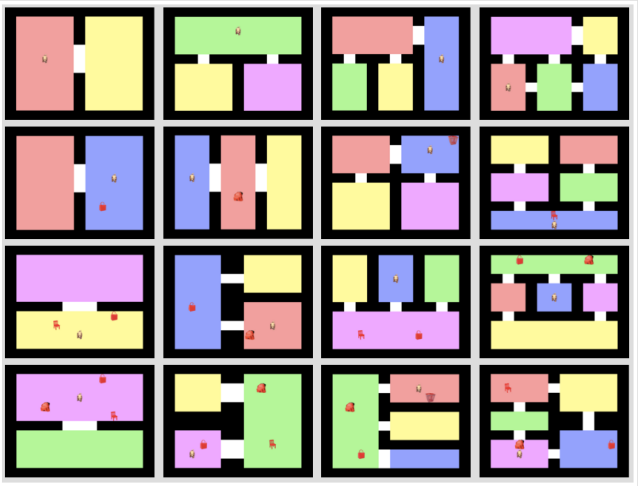

Figure 2: The library of 16 environments is organized by the number of rooms and objects. There is a command list for each environment.

training directly on the final task, M_t. In our setting, speed is measured via the number of trainer feedbacks required to learn.

In this paper, the 16 source tasks are provided.[1] Each task M_i is defined by 1) a training environment with an initial state and 2) a text command. The library of environments is shown in Figure 2. The 16 tasks are organized along two dimensions: the number of rooms and the number of moveable objects. For ease of description, we number the environments in the grid from 1 (top left) to 16 (bottom right) in English reading order. 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."

The target task command is "move the bag to the yellow room". It is not included for any environment to disallow training directly on the target command. Furthermore, the target task room layout is not in the set of 16 tasks used to construct curricula. To study the effect of the target task's complexity on the performance of curricula, we design two target task room layouts with the same command as shown in Figure 1 and Figure 3. It is worth noting that even though there are the same number of possible tasks the agent could execute in these two environments, the second target task is harder than the first one because there are more competing hypotheses on the agent's way to the goal state in the second target task.

4.3 Curriculum Learning

Using the probabilistic model and a curriculum, an iterative training regime proceeds as follows:

[1]Asking humans or agents to *construct* source tasks is an interesting problem left for future work.

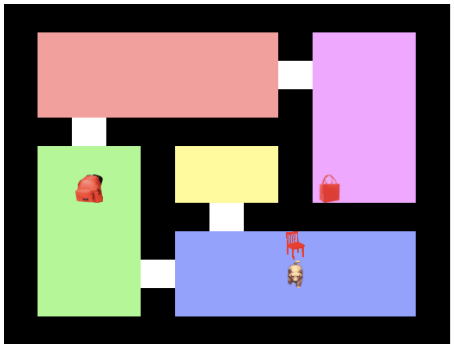

Figure 3: The target environment #2 (command: "move the bag to the yellow room") used in our study has a dog, five colored rooms and three objects (chair, bag, and backpack).

1. The agent receives an English command.
2. From this command, a distribution over the possible tasks for the current state of the environment is inferred using Bayesian techniques.
3. This task distribution is used as a prior for the goals in goal-directed SABL.
4. The SABL algorithm is trained for a series of time steps using reinforcement and/or punishment feedback given by an automated trainer.
5. After completing training[2], a new posterior distribution over tasks is induced and used to update the language model via weakly-supervised learning. After the language model is updated, training begins on the next task and command from the curriculum.

To study whether different methods for providing reward feedback to the agent as it learns within a curriculum influence which curricula are best, we consider three different automated trainers for step #4. We focus on "explicit" feedback, where a trainer provides positive or negative feedback, as a proxy for trainer effort. The *correct trainer* provides explicit feedback on 50% of the agent's actions and it is correct (reinforcement for actions consistent with optimal policy, punishment otherwise).The *error-prone trainer* also provides feedback on 50% of the agent's actions but when it provides feedback, it provides incorrect feedback 20% of the time[3] (and provides correct feedback 80% of the time). The *entropy-driven trainer* uses the entropy of the agent's policy to better target its feedback. This trainer provides feedback on 50% of the agent's actions if the entropy (H) of the agent's current action selection is high ($H > 0.1$) (i.e., the agent has high uncertainty in the optimal policy). The entropy of the action selection is used to summarize the agent's confidence:

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

where A is the set of possible actions, F is the history of feedback events from the trainer, and $\Pr(a = a^*|s, F)$ is the probability that action a given state s and feedback history

[2]Training for a task is completed once the agent stops at the goal state.

[3]Previous work in a similar setting found that a human trainer's error rate was roughly 20%.

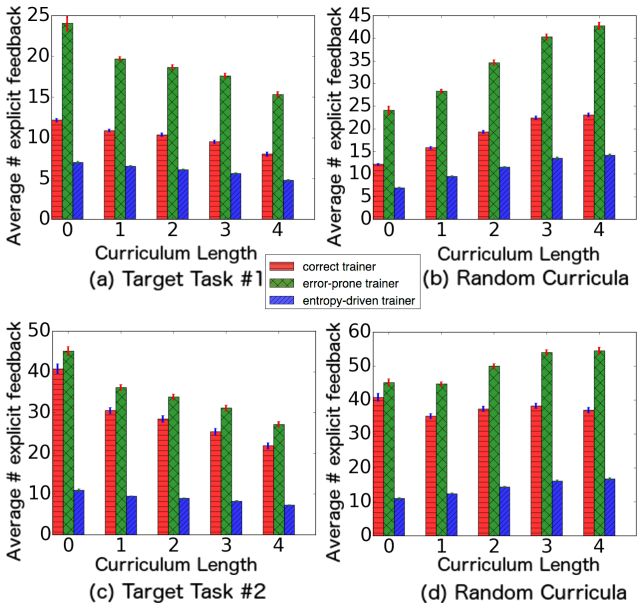

Figure 4: Average feedback needed to learn (a) the target task #1, (b) all tasks (including the target task #1), (c) the target task #2, or (d) all tasks (including the target task #2) on four sets of random curricula (or no curricula) with different automated trainers.

F is the optimal action (a^*). If the trainer provides feedback, the feedback is correct. Actions with $H \leq 0.1$ never receive feedback[4].

5. SIMULATION RESULTS

In this section, we analyze how curricula affect the number of trainer feedbacks required to learn.

5.1 Curriculum Effects

We hypothesized that 1) curricula could reduce the amount of feedback required to learn, 2) longer curricula would reduce the feedback required more than shorter curricula, and 3) feedback required could be reduced more as the target task's complexity increases. We generated four sets of random curricula of lengths $n = \{1, 2, 3, 4\}$. There were 200 curricula for each of the four sets. Each curriculum was generated by randomly selecting a sequence of the 16 environments and corresponding commands from the grid, allowing repeats. It is worth to 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 trainers and compared to directly learning the target task. For each of the two target tasks (shown in Figure 1 and Figure 3), we recorded the average amount of feedback required to learn 1) in the target task, and 2) in all tasks within the curricula (including the target task). Figure 4 summarizes these results. Error bars show standard er-

[4]When $H \leq 0.1$, the probability of the most likely task is $> 99\%$, indicating a near zero-probability of an incorrect action being taken. Trainer feedback would not help.

rors. As we expected, compared to directly learning each of the two target tasks, all four sets of random curricula could reduce the amount of feedback required to learn (shown in Figure 4(a) and Figure 4(c)). Feedback required could be reduced more in the second, harder target task than in the first, demonstrating that more benefits of curricula could be found as the target task's complexity increases. We also find that longer curricula always reduce the feedback required more than shorter curricula in both target tasks. However, in the harder target task, longer curricula do not always result in more feedback in total required than shorter curricula when accounting for the feedback spent learning the curriculum (shown in Figure 4(d)). This demonstrates that longer curricula do not always result in worse agent performance in learning all tasks within the curricula (including the target task). It is our expectation that the type of automated trainer used does not change these results—the method for providing feedback does not change the relative quality of the curriculum along these metrics.

Recall that a more difficult goal of curriculum design is to construct a sequence such that training on the entire curriculum and final task is faster than training directly on the final task. As shown in Figure 4(b), for the first target task, none of the four sets of random curricula result in a lower total amount of feedback required. However, Figure 4(d) shows that for the harder target task, all four sets of random curricula could result in a lower total amount of feedback required relative to directly learning the target task under the correct trainer, which achieves the more difficult goal of curriculum design. Unpaired two sample t-test shows that this difference was statistically significant ($p \ll 0.01$). It implies that as the target task's complexity increases, we could find more curricula resulting in faster total training time, while improving the agent's learning performance in the target task. However, even for the harder target task, almost no curricula result in a lower total amount of feedback required under the error-prone trainer or entropy-driven trainer. We believe the probability of receiving wrong feedback from the error-prone trainer makes it more difficult for the agent to fully leverage what it has learned from each task in the curricula. For the entropy-driven trainer, the number of trainer feedbacks was minimized compared to the other two trainers, which makes it harder to improve.

A two-way ANOVA test on the results in Figure 4(a) and Figure 4(c) shows that differences in the amount of feedback required for the agent to successfully learn each of the two intended tasks between the three automated trainers or four sets of random curricula were both statistically significant ($p \ll 0.01$), verifying that different curricula can have substantial impact on training speeds. The interaction effects of automated trainer and curriculum length on curriculum quality achieved were statistically significant ($p < 0.05$). Simple main effects analysis showed that the feedback differences between four sets of random curricula were significant within each of the three trainer groups.

5.2 Transition Dynamics

When selecting tasks for a curriculum, designers can construct task sequences that introduce complexity in certain ways, which we refer to as curriculum transition dynamics. We are interested in studying which transition type(s) is the best or worst for minimizing the amount of feedback required for the agent to learn the target task, or all tasks within the

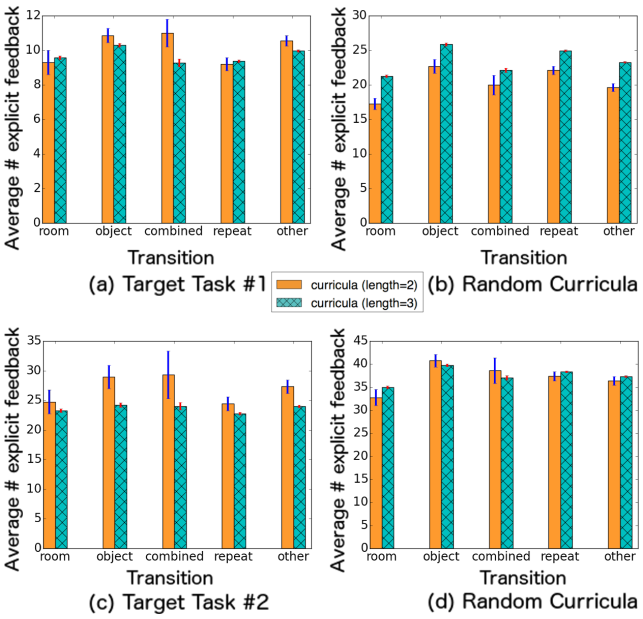

Figure 5: Average feedback needed to learn (a) the target task #1, (b) all tasks (including the target task #1), (C) the target task #2, or (d) all tasks (including the target task #2) on curricula with length 2 or curricula with length 3 under the correct trainer.

curriculum. For the 4×4 grid (shown in Figure 2), we defined five different ways to change the environment complexity when designing a curriculum: room transition, object transition, combined transition, repeat transition and others. For a given task M_i in a curriculum, a transition to M_{i+1} is a *room transition* if and only if the number of rooms increases between M_i and M_{i+1}. If the number of objects increases, it is an *object transition*, and if they both increase it is a *combined transition*. If $M_i = M_{i+1}$, it is a repeat "transition." All other cases are considered as *other transitions*.

To find which transition types result in the best agent performance, we chose environments 1, 3, 9 and 11 (four environments with varying numbers of rooms and objects) and evaluated all possible curricula with lengths 2 and 3 for both target tasks, using the correct trainer. Figure 5 summarizes these results. For curricula with length 2, we find that for both target tasks, 1) room transitions were the best (or second best) for minimizing the feedback required to learn all tasks within the curricula (or the target task), and 2) combined transitions were the worst for minimizing the feedback required in the target task, while object transitions were the worst for minimizing total feedback required in all tasks within the curricula. For curricula with length 3, for both target tasks, room transitions were the best for minimizing total feedback required in all tasks, while object transitions were the worst on both evaluation metrics. This suggests that curricula that follow the room transitions result in considerably better agent performance—in learning both the target task and all tasks within the curricula—compared to the curricula that make use of any other transition types.

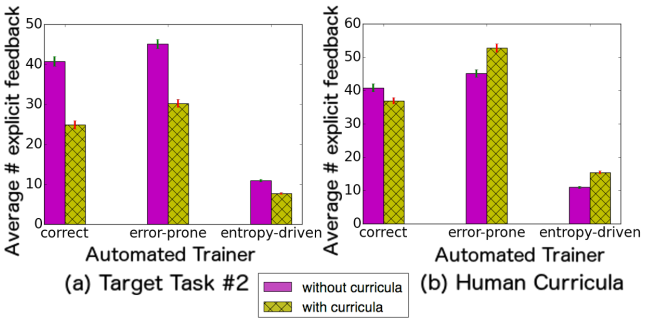

Figure 6: Average feedback needed to learn (a) the target task #2 or (b) all tasks (including the target task #2) with and without human-designed curricula.

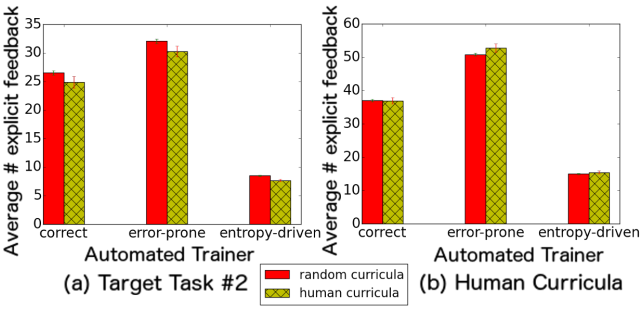

Figure 7: Average feedback needed to learn (a) the target task #2 or (b) all tasks (including the target task) with random and human-designed curricula.

6. HUMAN SUBJECTS RESULTS

To study whether non-expert humans (*i.e.*, workers on Amazon Mechanical Turk) can design good curricula for an agent, we developed an empirical study in which participants were asked to design a set of training assignments for the dog to help it quickly learn to complete the final target assignment.

6.1 Design

First, participants had to pass a color blindness test. Second, participants filled out a background survey. Third, participants were taken through a tutorial that 1) walked them through two examples of the dog being trained to help them understand how the dog learns to complete a novel command successfully using reinforcement and punishment feedback, and 2) taught them how to use the interface to design a curriculum for the dog. Participants were told that 1) their goal is to design a set of assignments for the dog to train on such that the dog can quickly learn to complete the final target assignment, 2) they can observe the whole process of the dog being trained in each assignment in their designed curriculum and the target assignment, and 3) higher payment would be given to them if a better curriculum was designed.

Following the tutorial, participants selected environments and commands from the 16-environment grid (Figure 2) in any order to design their own curricula. The target task (Figure 3) is shown on the right side of the screen to remind participants of the goal for the designed curricula. We choose the second target task to better explore whether non-expert humans can design good curricula for improving agent learning in the harder final task. Upon finishing designing a curriculum (containing at least one task), participants could watch the automated (correct) trainer teach the entire curriculum. Participants were required to redesign the curriculum at least once.

6.2 Experimental Setup

To study the effect of the ordering of source tasks on human performance in designing curricula, we varied the order of the 16 environments in the grid. We transposed the grid, swapping Environments 1 and 16, 2 and 12, *etc.*, such that the difficulty level of the environments gradually decreases from left to right, and top to bottom. Participants were randomly assigned to an experimental condition:

- **Gradually Complex**: the number of rooms increases from left to right, and the number of objects increases from top to bottom.
- **Gradually Simple**: the number of rooms decreases from top to bottom, and the number of objects decreases from left to right.

6.3 Results

This section summarizes our user-study results. Our experiment was published on Amazon Mechanical Turk as a set of Human Intelligence Tasks. We considered data from 80 unique workers, after excluding 15 responses who we identified as users who just submitted as fast as possible so as to be paid. We identified such users as those whose completion time was shorter than 5 minutes (the average completion time was 22 minutes 18 seconds, with a standard deviation of 8.3 minutes) or if both designed curricula contained only a single task. There were 40 participants for each experimental condition (gradually complex and gradually simple). Each participant only saw one grid type when designing curricula.

6.3.1 Participant Performance

Recall that the objective of the curriculum design problem in sequential decision tasks is to design a curriculum the agent would train on such that the agent could successfully complete its target task quickly and with little feedback. Therefore, we first examined whether users could successfully design such curricula by computing the average amount of feedback needed for the agent to learn the target task after being trained on all initial and final curricula designed by them. Recall that all users had to provide at least two curricula, but we evaluated only the initial and final curricula. Each curriculum was evaluated 20 times.

Figure 6(a) shows that, compared to directly learning the target task #2, less feedback was required for the agent to master the intended task after training on curricula, under all three automated trainers. Figure 6(b) shows that less feedback in total was required for the agent to learn all tasks within the curricula (including the target task #2) than only learning the target task under the correct trainer, which achieves the more difficult goal of curriculum learning. It is also worth noting that participants were not given any feedback on the quality of the curricula they created. A two-way ANOVA test shows that the differences in the amount of feedback required to learn the target task between us-

ing the curricula or not using the curricula were statistically significant ($p \ll 0.01$). The feedback differences between the three automated trainers were also statistically significant ($p \ll 0.01$). The interaction effects of these two factors on curriculum quality achieved were statistically significant ($p < 0.05$). Simple main effects analysis showed that significantly less feedback was required for the agent to master the intended task after training on curricula than learning from scratch within each of the three trainer groups.

To study whether non-expert humans can design good curricula, we compared the average amount of feedback needed for the agent to learn 1) the target task, and 2) all tasks (including the target task) after being trained on all initial and final curricula designed by humans and on all four sets of random curricula (shown in Figure 7). The result shows that human-designed curricula result in 1) less feedback required for the agent to master the intended task, and 2) more feedback in total required for the agent to learn all tasks (including the target task) than random curricula, demonstrating that non-expert humans are good at designing curricula in terms of improving the agent performance in learning the target task in this sequential decision task domain.

6.3.2 *Transition Dynamics*

We then examined the overall quality of the curricula relative to those we found during our previous exploration of machine-generated curricula.

We analyzed the most popular transitions in the two experimental conditions by computing the frequency of each of five transitions being followed for each environment. We found that 1) the room transition was the most-frequently used transition in the gradually complex condition, which was the best transition type we found before for curricula with length 2 or 3 in both target tasks, 2) the other transition was the most-frequently used transition in the gradually simple condition, which was shown to be better than object or combined transition for curricula with length 2 or 3 in both target tasks, and 3) the combined transition was the least-frequently used transition in both experimental conditions, which was the worst for minimizing the feedback required in the target task for curricula with length 2 in both target tasks. Thus, non-expert users can generate efficient curricula that include useful transition types. A Chi-squared test showed that the transition types used between the two experimental conditions were significantly different ($p \ll 0.01$), demonstrating that the ordering of source environments affected the way participants chose to design curricula.

We then examined the overall agent performance in learning both the target task and all tasks within the curricula in these two experimental conditions. We found that the curricula designed in these two experimental conditions result in similar agent performance on these two evaluation metrics, demonstrating that the ordering of source environments does not affect human performance in designing curricula for the agent to train on.

6.3.3 *Environment Preference*

We hypothesized that some source environments in the grid would be preferred by users when designing their curricula. Analyzing the properties of these environments might enrich the general principles regarding efficient curricula and

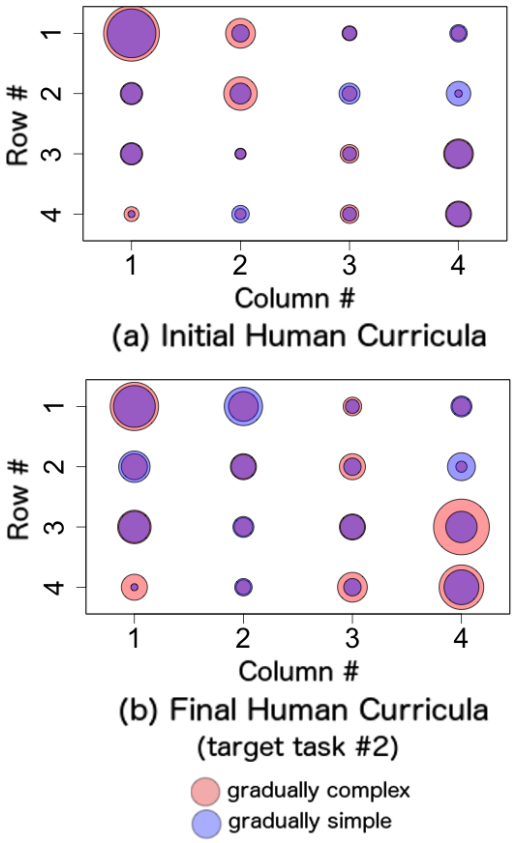

Figure 8: The probability of each environment being included in a human-generated curriculum from both conditions. The purple circle represents the overlap.

inspire the development of new machine-learning algorithms that accommodate human teaching strategies. Therefore, we explored user preference in each environment by computing the fraction of users who selected corresponding environments at least once.

Figure 8 summarizes user preference in each of the 16 environments when designing initial or final curricula in two experimental conditions. The environments are shown in the same order as the gradually complex condition. A larger dot represents a higher probability of the corresponding environment being chosen. We find that when designing initial curricula, users 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 users preferred to choose 1) 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, and 2) more complex environments that are more similar to the target environment. (Environments 12 and 16 are two of the most similar ones to the target environment.)

We also note that users had a similar probability of choosing the two simplest environments (1 and 2) after varying the order of the 16 environments. Fisher's exact test shows

that the frequency of each of the 16 environments being selected by users for initial or final curricula was not significantly different ($p > 0.05$) between the two experimental conditions, suggesting that the ordering of source environments does not influence participants' preference in choosing environments. We believe that savvy users 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 a new type of complexity. These principles can be highly useful for the design of new machine-learning algorithms that accommodate human teaching strategies.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced and defined the curriculum design problem in the context of sequential decision tasks, where the goal is 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) the choice of curriculum can have substantial impact on training speeds while longer curricula do not always result in worse agent performance 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 method for providing reward feedback to the agent as it learns within a curriculum does not change which curricula are best. We also present an empirical study designed to explore how non-expert humans generate such curricula. We demonstrated that non-expert users can 1) successfully design curricula that result in better overall agent performance than learning from scratch, even in the absence of feedback on their quality, and 2) discover and follow some salient patterns when selecting and sequencing environments in the curricula—an attribute we plan to leverage in the design of RL algorithms in the future.

Considering that the tasks in real world could 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 the training time is an important performance metric. Second, we could incorporate the salient principles (e.g., isolating complexity) we find about humans when designing curricula into the automatic process of generating useful source tasks in any task domain. We could also build new learning algorithms with inductive biases that favor the types of curricular changes that human trainers tend to use. Finally, the interface design could be improved to guide the non-experts to design better curricula.

Future work will study curriculum design 1) when users can create a sequence of novel source tasks for the agent to train on, and 2) when users can see a score of the designed curricula and use this feedback in their design process, and 3) when the learning algorithm is able to leverage 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 grants AFRL FA8750-14-1-0069, AFRL FA8750-14-1-0070, NSF IIS-1149917, NSF IIS-1319412, USDA 2014-67021-22174, and a Google Research Award. CIIGAR research is supported in part by NSF grant IIS-1319305.

REFERENCES

[1] A. Baranes and P.-Y. Oudeyer. Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61(1):49–73, 2013.

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

[3] 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*, 16(2):79–85, 1990.

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

[5] S. Griffith, K. Subramanian, J. Scholz, C. Isbell, and A. L. Thomaz. Policy shaping: Integrating human feedback with reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2625–2633, 2013.

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

[7] W. B. Knox and P. Stone. Interactively shaping agents via human reinforcement: The TAMER framework. In *The Fifth International Conference on Knowledge Capture*, September 2009.

[8] 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*, pages 475–482, 2012.

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

[10] 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*, pages 1721–1728, 2011.

[11] 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. *Journal of Autonomous Agents and Multi-Agent Systems*, pages 1–30, 2015.

[12] 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.

[13] L. Mihalkova and R. J. Mooney. Using active relocation to aid reinforcement learning. In *FLAIRS Conference*, pages 580–585, 2006.

[14] 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.

[15] A. Y. Ng, D. Harada, and S. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. 1999.

[16] B. F. Skinner. Reinforcement today. *American Psychologist*, 13(3):94, 1958.

[17] H. B. Suay and S. Chernova. Effect of human guidance and state space size on interactive reinforcement learning. In *2011 Ro-Man*, pages 1–6, 2011.

[18] K. Subramanian, C. L. Isbell Jr, and A. L. Thomaz. Exploration from demonstration for interactive reinforcement learning. In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems*, pages 447–456, 2016.

[19] R. S. Sutton and A. G. Barto. *Introduction to reinforcement learning*, volume 135. MIT Press Cambridge, 1998.

[20] 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*, pages 871–878, 2007.

[21] 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.

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

[23] M. E. Taylor, P. Stone, and Y. Liu. Transfer Learning via Inter-Task Mappings for Temporal Difference Learning. *Journal of Machine Learning Research*, 8(1):2125–2167, 2007.

[24] A. L. Thomaz and C. Breazeal. Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In *AAAI*, volume 6, pages 1000–1005, 2006.

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

[26] C. M. Vigorito and A. G. Barto. Intrinsically motivated hierarchical skill learning in structured environments. *IEEE Transactions on Autonomous Mental Development*, 2(2):132–143, 2010.

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

[28] 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*, pages 1015–1022, 2007.