Incorporating Human Prior Knowledge to Reinforcement Learning Agent for Atari Games

Wenbo Song (ws1542@nyu.edu)
Hao-Ning Wu (haoning.wu@nyu.edu)
Rohan Raj (rohanraj@nyu.edu)
Carl Barbee (crb616@nyu.edu)
Courant Institute of Mathematical Sciences,
New York University, New York, NY 10003

Abstract

Humans have shown a remarkable ability to acquire and utilize new information in a diverse set of environments, whereas reinforcement learning agents struggle to adapt, while suffering an exorbitant training cost and poor generalization in unseen environments. Therefore, it's essential to incorporate the concept of world knowledge into an agent to create a robust, general-purpose RL model. In this project, we attempt to equip the agent with the recognition of basic components of an Atari game environment through curriculum learning—gleaned from human developmental psychology—and evaluate its performance. Our best agent was pre-trained on a carefully designed curriculum to learn to complete a new game 5x faster than regular agents. By analyzing our agent's behavior, we can examine how it acquires knowledge and skills differently from humans.
Keywords: Human priors, reinforcement learning, Atari games

1 Introduction

Humans possess a wealth of world knowledge that they can apply to a variety of tasks. Previous research has shown that humans are capable of utilizing different forms of information such as visual, auditory and kinesthetic (Dubey et al. 2018; Tsividis et al. 2017; Lake et al. 2017) to tackle new tasks within a short time frame and only a few examples. This is easily seen in complex environments like video games. The possession of various world knowledge and the ability to exploit it enable humans to significantly outperform the majority of the state-of-art machine learning algorithms in the domain of learning and decision-making in multiple problem settings (Muggleton et al. 1989; Kattan, Adams, and Parks 1993; Lake et al. 2017).

Meanwhile, powered by the advances in deep learning and computing power, we have seen many great successes in machine learning applied to modeling a wide range of problems such as control (Schmidhuber 2015), robotics (Kober, Bagnell, and Peters 2013) and games (Mnih, Kavukcuoglu, Silver, Rusu, et al. 2015; Silver et al. 2016; Tesauro 1995). In particular, reinforcement learning has differentiated itself from supervised and unsupervised learning by being interactive with the environment (Sutton and Barto 2018). However, the high profile success of reinforcement learning algorithms requires super-human level computation and—what's even worse—they are vulnerable to subtle modifications of the environment (Lake et al. 2017). Fortunately, research in cognitive science has shed some light on potential improvements for reinforcement learning models. Lake et al. 2017 argue that in order for agents to truly be intelligent they must possess an intuition and understanding of their environment that is currently lacking in machine learning models.

It's worth mentioning that humans are not born with a clean slate, but instead possess *start-up software* which helps them achieve their goals and is further built upon throughout their life (Wellman and Gelman 1992). For a person to master a language it would require many years of training in a highly organized and deliberately designed curriculum. This learning from a curriculum is observed in multiple human learning scenarios, and is essential in concept acquisition(Rohde and Plaut 1999). Previous studies at the intersection of cognitive science and machine learning have argued that this learning paradigm, known as curriculum learning, can also be applied to machine learning algorithms as an effective approach to improve sample efficiency and asymptotic performance (Portelas et al. 2020; Bengio et al. 2009).

In this paper, we're interested in extracting the salient aspects of an Atari game environment, that make the average person capable of completing the game within minutes, and teaching the RL agent these concepts to hopefully improve the performance. Concretely, our approach is to, first, decompose a complex Atari game into a series of simplified environments with a limited number of objects, then pre-train the agent using this newly-designed curriculum. By comparison to an agent without curriculum pre-training, our approach can speed up a RL agent's training on unseen maps by 5x. The results suggest that learning a lower-level concept first enables the agent to accomplish similar tasks more efficiently.

2 Prior Knowledge

The cognitive science literature contains considerable research on human's world knowledge including the ability to predict, acquire, and utilize new information. Humans are capable of synthesizing visual, auditory, and kinesthetic information and applying it to inform their future actions by visualizing potential outcomes (Lake et al. 2017; Tsividis et al. 2017). Many studies have shown that humans have developmental *start-up*

software which is essential to their ability to learn from their experiences and interact with their environment (Wellman and Gelman 1992), specifically, the theory of intuitive physics and psychology. *Intuitive physics* is the notion that humans have primitive object concepts that allow them to filter out illogical behavior. For example, a child can understand that instead of a ball teleporting from point A to B it will instead follow a natural trajectory. *Intuitive psychology* is the idea that other people have goals and beliefs which constrain their learning and predictions. For instance, if a child watches another person play a game, then they can infer that the agent is seeking a reward while avoiding punishment. This type of thinking constrains or eliminates other implausible strategies (Lake et al. 2017).

We found these theories to be essential in determining human performance on games that are consistent with reality. For example, we are not interested in games where there's a probability that pressing up on a keyboard will make the person's avatar go down, since this doesn't conform to the real world. Tsividis et al. 2017 demonstrated that humans are capable of performing well on most Atari games within a matter of minutes. Furthermore, they found Atari to be a compelling test bed for comparing human and agent performance. In their study, they compared the performance of human learning trajectories for several Atari games and tested a series of hypotheses to determine why humans possess such rapid learning. The researchers found that humans utilize intuitive theories which allow them to generalize from a few examples.

However, human prior knowledge can come at a cost. In Dubey et al. 2018 and other studies (Doshi-Velez and Ghahramani 2011; Kulkarni et al. 2016) a person's world knowledge might be a deficit in partially observable environments or dealing with randomness (Gureckis and Love 2009), but they do seem to perform well in fully observable spaces where the environment is similar to the real world (Lake et al. 2017), such as Atari games (Tsividis et al. 2017). We are interested in providing a human's world knowledge—climbing ladders, avoiding fire—to an agent to determine if these pieces of information will help it become more robust to new environments. These intuitive mechanics are essential to a human's ability to interact with their environment and adapt quickly. Our experiments aim to imbue this knowledge into our RL agent and examine its performance to determine how well the agent will perform.

3 Curriculum Learning

Studies from cognitive science have shown that humans have the ability of exploiting the related knowledge learned previously to ease the learning process of new concepts. Put differently, a person's acquisition of a new complex concept or behaviour can be assisted by the external guidance of a sequence of simplified components (Peterson 2004). This learning paradigm is observed in numerous animal behaviour experiments, and is widely acknowledged as *shaping* in cognitive science, that dates back to Skinner 1938. The basic idea of *shaping* is to start with easy intermediate sub-tasks and gradually upgrade the difficulty to finally approach the original complex task. For example, in language acquisition, a beginner wouldn't start with attempting to understand a complicated structured sentence. On the contrary, a fundamental understanding of the alphabet, pronunciation, simple words, and grammar would be a much smarter place to start and is probably the most logical method as suggested in (Elman 1993).

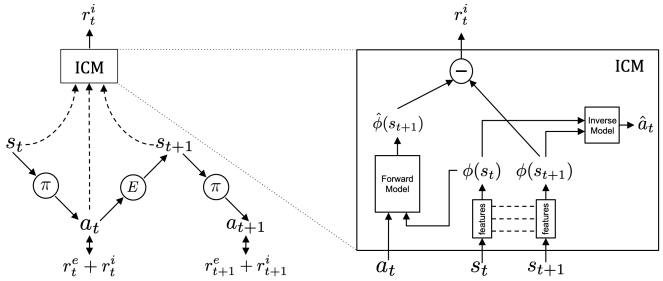


Figure 1: A3C-ICM, taken from (Pathak et al. 2017)

A similar methodology has also been actively evolved in the community of reinforcement learning, known as curriculum learning. Enlightened by the success of teaching a recurrent network grammatical expression by starting small (Elman 1993), curriculum learning has been proved to be an effective training strategy for reinforcement learning algorithms in various problem settings, such as video games (Wu and Tian 2016; Svetlik et al. 2017), navigation (Florensa et al. 2017) and robotic control (Dorigo and Colombetti 1998; Justesen and Risi 2018). The application of reinforcement learning algorithms in real-world problems always suffers the difficulty of convergence in large-scale, complex environments with a high-dimensional observation space. However, in curriculum learning, the convergence will be guided through a sequence of pre-training in simpler environment settings.

In this paper, curriculum learning is used as an approach to teach an agent human world knowledge of the Atari game. More specifically, to teach the agent that fire is "bad", a group of maps with different levels of multiple individual/consecutive fire pits are used in the pre-train phase. Similarly, a group of maps with ladders are used to pre-train the model to *understand* the concept of ladder and the behaviour of climbing.

4 Reinforcement Learning Model:A3C-ICM

To further investigate the feasibility of incorporating human world knowledge into a reinforcement learning agent via curriculum learning and whether this approach is effective for improving the performance, the Asynchronous Advantage Actor Critic with intrinsic curiosity module(A3C-ICM) proposed by (Pathak et al. 2017) is used in this paper.

The intrinsic reward module of A3C-ICM provides a curiosity-driven reward and a policy that attempts to maximize this reward. As illustrated in Fig.1, the agent learns the optimal policy by optimizing the sum of the extrinsic rewards r^e from the environment and curiosity based intrinsic rewards r^i generated by Intrinsic Curiosity Module (ICM). Within ICM, the states S_t and S_{t+1} are firstly used to predict action

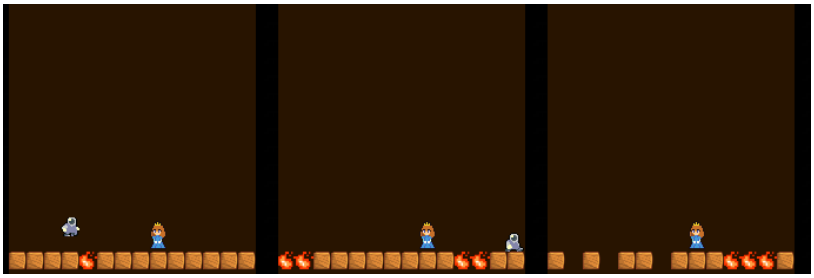


Figure 2: Fire Concept Maps

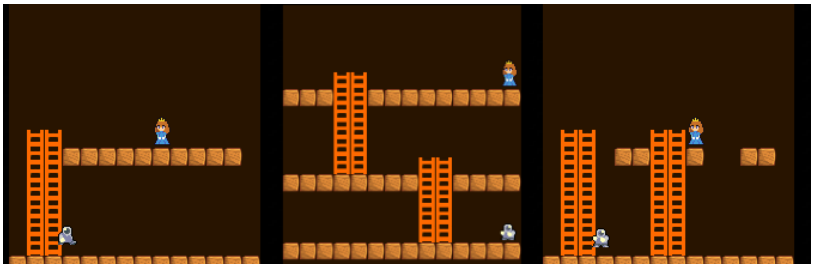


Figure 3: Ladder Concept Maps

$\hat{a_t}$ using inverse model after encoded as $\Phi(S_t)$ and $\Phi(S_{t+1})$,

$$\hat{a_t} = g(s_t, s_{t+1}; \theta_I)$$

where, the network parameter θ_I requires training to optimize with the loss function:

$$min_{\theta_I} L_I(\hat{a_t}, a_t)$$

Then, $\Phi(S_t)$ and a_t are fed into the forward model to predict the feature representation $\Phi(\hat{S_{t+1}})$,

$$\Phi(\hat{S_{t+1}}) = f(\Phi(s_t), a_t; \theta_F)$$

where, θ_F is the network parameter for the forward model that are optimized by the loss function L_F:

$$L_F(\Phi(s_t), \Phi(\hat{S_{t+1}})) = \frac{1}{2} \left\| \Phi(\hat{S_{t+1}}) - \Phi(s_t) \right\|_2^2$$

Therefore, the intrinsic reward signal r_t^i is the prediction error of $\Phi(\hat{S_{t+1}})$ and $\Phi(S_{t+1})$:

$$r_t^i = \frac{\mu}{2} \left\| \Phi(\hat{S_{t+1}}) - \Phi(s_t) \right\|_2^2$$

where μ is scaling factor.

The goal of the agent becomes to train a policy π that maximizes the expected total reward of extrinsic and intrinsic reward $r_t = r_t^e + r_t^i$,

$$\max_{\theta_P} \mathbb{E}_{\pi(s_t, \theta_P)}[\sum_t r_t]$$

where θ_P represents the policy parameter.

In an environment with loose extrinsic rewards like Atari games, the curiosity module can serve as an intrinsic reward signal to help the A3C agent's exploration strategy regardless of the disturbance from the environment. The study from Pathak et al. 2017 has shown that A3C-ICM can learn a better final optimal policy with less time than regular A3C agents in these various sparse reward environments.

Therefore, in our experiment, we train all of the A3C agents with ICM, which guarantees the basic convergence of the RL agent in the game environment we'll be using. Building on this, our own experiments are designed to incorporate human world knowledge into the RL model, and further evaluate how this information assists an agent in improving its learning performance.

5 Experiments

Atari games are generally used as a proper simulation for continuous decision-making problems in the real world. In this paper, the game, Monster Kong, is selected as the foundation for our experiments. In this game, the avatar can perform actions selected from the action space: $\{left, right, up, down, jump, stay\}$. The objective of the game is to avoid the obstacles and reach the princess.

Simple as it may seem at first glance, it requires players to have the recognition of avoiding the fire pits by jumping, climbing to a higher level using ladders and rescuing the princess, which are the major components that the RL agent is expected to learn via curriculum learning in our experiments.

5.1 Environment Design

We designed a series of *concept maps*, each of which contains a specific concept we want the agents to learn. Our motivation for designing these maps was an incrementally difficult curriculum we determined was necessary. For example, in order to make the agent learn that fire is "bad" we design several levels of fire maps (Fig. 2):

1. Levels with fire pits and a goal state to teach the agent to dodge fire pits irrespective of their location.

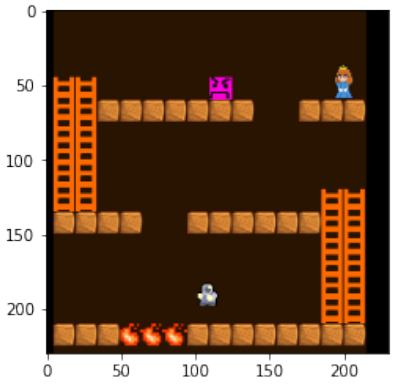


Figure 4: Target Map

2. Levels with multiple consecutive fire pits and a goal state to introduce the concept of consecutive fire pits.
3. A combination of the above 2 with gaps in the floor to expose the agent to jumping over fire pits and gaps.

The other set of experiments focus on teaching the agent to climb (Fig. 3):

1. Levels with a single ladder for teaching the concept of climbing.
2. Levels with multiple ladders to expose the agent to ascending/descending ladders and, hopefully, choosing the shorter path.
3. Finally, a combination of the two with gaps in the floor to teach the agent to jump and climb.

5.2 Training Details

Our *expert models* are pre-trained on 5 base concept maps that fall under a category (e.g. ladders, fire pits, etc.). After the pre-training phase, we fine-tune the expert model on another *target map*. Lastly, we compare the learning speed between the expert agents and a *baseline model* which is trained from scratch on the target map.

Hyper-parameter setting: We follow the training and pre-processing procedure described in (Pathak et al. 2017) for our baseline models and expert models. Actions are repeated 4 times during training and once during inference (Mnih, Kavukcuoglu, Silver, Graves, et al. 2013). 16 agents were trained asynchronously using ADAM optimizer on 4 AMD Opteron 6272 (2.1 GHz) CPUs with 64 cores. We use a learning rate of $1e-4$. The maximum number of actions performed during exploration phase in each episode is set to 500. Our best performing model uses the sparse reward setting where the agent only receives a +1 reward when it reaches the goal state.

Map randomization: During a training episode, our agents randomly picked one concept map in the same category. To avoid over-fitting so the agent does not memorize actions and is able to generalize well, we try the following techniques:

1. Random Player Position (RPP): The player always starts at the bottom level and needs to reach the princess at the top level. The starting position of the player is randomly chosen from one of predefined positions in order to teach the model to climb up.
2. Switching Player and Princess (SPR): Randomly switch the starting positions of the princess and the player. We want to teach the model to climb up and down.
3. Flipping the Maps (FM): Randomly flip the map horizontally.

Model evaluation: To select the best model from all checkpoints, we evaluate the game success rates on training maps. We claim that using a non-greedy policy (i.e., sample an action from the output probability distribution) can better reflect the model's capability. If we use a greedy policy (i.e., select the action with maximum probability), a common-mode failure is that a slight displacement of the player causes it to get stuck indefinitely. We've also tried to evaluate on test maps that were not used during training. Nevertheless, no matter how we trained the models, very few of them could complete any of the test maps. Further work is needed to explore generalizing to more complex, unseen environments.

5.3 Results

Hyper-parameter tuning: The comparison between some of our hyper-parameters is shown in Fig. 5, among which the one with the best performance on the target map is selected as our final setting. In Pathak et al. 2017, they set the maximum step number during exploration to roughly ten times the optimal number of steps, which is usually below 30 for our maps. However, we have a larger action space—6 possible actions. If the value of this parameter is too small, the chance of the agent reaching the goal state would be too low. Whereas, using a value too large would make the agent collect a bunch of useless experience, resulting in a longer time to converge.

Moreover, we found that using a denser reward did not help the convergence. Our interpretation is that this is due to the structure of the maps. In our target map, a monster is placed in front of the princess, so during training the agent will encounter it each time it attempts to complete the map. If the agent is hurt by the monster, it will receive a negative reward, that can cause a low estimate for that state. Therefore, we claim that the curiosity model encounters a boredom issue (Pathak et al. 2017) leading it to drift towards other states. From Fig. 5, we observe that the agent can get a total return of 0 after sufficient training, which suggests that it has learned to

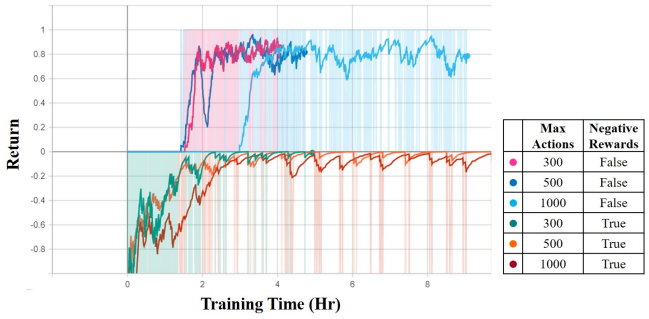


Figure 5: Return vs. Different Training Setups

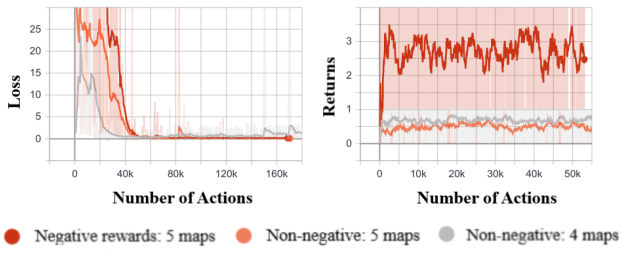


Figure 6: Negative Reward Structure

avoid danger. However, the agent fails to reach the princess and receive a reward of 1.

The similar results are observed in the fire concept maps. To overcome this issue, we tried to give a higher reward on reaching the princess based on the hypothesis that it would incentivize the agent to get past the enemies. However, as shown in Fig. 6, this reward structure resulted in erratic returns and the convergence was also slower than its counterpart.

Randomized training for expert models: Table 1 is a summary of the different techniques of randomization used in our experiments. Three models are created by incrementally adding a technique described earlier. As expected, the more random our training data, the longer it took to converge to its best performance. Moreover, the most random setting (+FM) failed to complete the training maps a 100% of the time. This suggests there might be some under-fitting problem. It would be interesting to increase the model capacity in the future.

Model	Effective # Maps	# Training Steps (M)	Success Rate
RPP	5	8	100%
+SPR	10	14.4	100%
+FM	20	14.4	90%

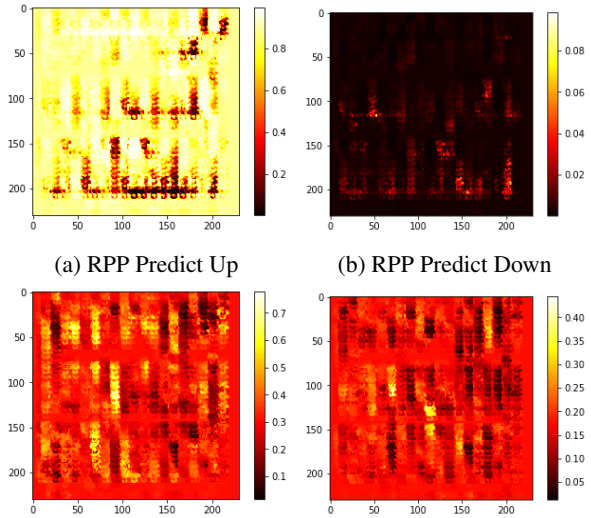
Table 1: Pre-training the expert models

Training on target maps: We compared the efficiencies of expert models and baseline models by their convergence times on the target map in Table. 2. The results are averaged over 5 runs. The convergence time is defined as the time that the model needed to reach a running average of 80% success rate on the map. After this point, the success rate of every model starts oscillating between 70% and 90%.

Model	# Training Steps (M)
Baseline	4.515±2.51
RPP	**0.896±0.32**
+SPR	1.108±0.70
+FM	1.139±0.75

Table 2: Target Map Result

Our expert models always converged faster than the baseline model. Surprisingly, using fewer randomization techniques resulted in better performance. Especially, the RPP model converges roughly 5x times faster than the baseline. However,



(a) RPP Predict Up (b) RPP Predict Down

(c) +SPR Predict Up (d) +SPR Predict Down

Figure 7: Heat Maps for Action Probability

differences between models are within one standard deviation, so the difference between the results are not statistical significant. Our explanation on such results is that the player learns too much redundant knowledge during the pre-training phase. That knowledge is useless in the target map where the player can achieve the goal without climbing down. Moreover, we observed that the training of the +FM model is unstable. First, its loss function fluctuates more sharply. Second, on a rare occasion after passing the 80% threshold, it suddenly broke down and was unable to complete the game anymore. Unfortunately, we haven't found an explanation to explain such a result.

Demystifying the behaviors of different models: In order to know what did our agent learn so that it can complete the target map much faster, we visualized the heat maps of the action probabilities in Fig. 7. The heat maps are generated by the following steps:

1. Place the player at one coordinate.
2. Run a forward pass of a pre-trained model to predict the probability of actions for the current scene.
3. Iterate the previous steps on all coordinates.

We omit the heat map for the +FM model because it shares similar patterns with the +SPR model's.

In Fig. 7a, the RPP model gives high probabilities for the "up" action at almost every positions. The only black regions align with the location with the platforms. One possible explanation is that when the agent gets close to the ladders, it jumps on it instead of walking to the bottom of the ladder and then climbs up, according to the observation on the generated videos of game plays. So, it is reasonable that the RPP model prefers going "up" when the player is in the air than on the ground.

In Fig. 7c,7d, the +SPR's heat maps show interleaving stripe patterns. We notice that the positions with the strongest responses have little correlation with where the ladders are. In-

stead, the higher response stripes appear at every other block. Note that due to the restriction of the game, the ladders must show up in pairs. Therefore, to find a ladder to climb up, the player only need to search half of the space. We conclude that the model didn't learn to identify the ladders, but learned to jump at intervals to increase its chance to grip the ladder. Lastly, the "up" heat map still possesses higher responses at most of the coordinates compared to the "down" heat map. This is probably because when the player wants to go down, it can jump right off the ladder instead of climbing down step by step. Therefore, the "down" action is only used in very few situations. Our above theories were verified by watching the generated videos.

6 Discussion

Human world knowledge is multifaceted and nuanced. Simply enumerating or pinpointing the specific aspects of how a person plays a game requires extraordinary effort and will inevitable exclude or simplify concepts. The results from our experiments suggest that incorporating prior information through a curriculum to train an agent is promising, but can have conflicting results depending on the information.

One major problem is that we assumed that fire and ladders would be essential parts to learn/utilize, but each had it's difficulties. Fire, we assumed, requires negative rewards to disincentivize the agent from walking/jumping into it, but this can lead to convergence challenges. Ladders should be straightforward, but the agent might work around them, e.g., jump off them when going down instead of climbing each rung. In future work it would be interesting to unpack our assumptions about what is and isn't essential to determine if an agent truly requires these pieces of information.

Another difficulty is to design a game accurately with one isolated concept. Even when there were only ladders, the concept maps still leaked a lot of information to the agent. For example, the platforms are walk-able and you can jump off a ladder at any point. Additionally, the agent might make use of other signals to help it navigate through the map. For instance, the agent might be learning to find the princess instead of learning how to utilize the ladder. It's unclear how to design a set of maps that provide exact world knowledge only about ladders.

When humans try to solve such games for the first time, they have a clear idea of how the actions should be used. This helps cut down on a lot of unnecessary exploration time. On the contrary, our agents developed efficient and reasonable strategies for these kinds of games and showed generalizability. However, we cannot jump to the conclusion that our agent truly learned to utilize ladders. For example, if there happens to be enemies in the middle of the air, it could kill our agent quite easily and make it difficult for the agent to learn. We believe there are still many aspects of human world knowledge that are essential to map-design for a more robust model.

In our experiments, to implicitly impart prior knowledge to an RL agent, we aimed to teach the agent to perform individual tasks like climbing ladders. However, the human psyche has a number of such models which work independently and cooperatively to make decisions. In order to mimic human psychology, we claim that using a similar approach (Tan 1993) will further help the agent to converge faster and be also able to generalize. These multi-agents can be thought of as simple building blocks assembled in a way to complete a much harder task. We believe this is a more robust way of training an agent because we're able to reach our goal in fewer actions, indicating that the agent isn't memorizing the map. Moreover, we can use these "general" building blocks in various combinations (Tan 1993) to achieve different goals, thereby increasing generalizability. This approach can be used during training or testing. For example, during testing, we can add the policies of 2 trained agents and pick the action which has the maximum value function. For our environment, this may help avoid the scenario where the ladder agent encourages the agent to go towards a fire pit, but the fire agent pulls the agent away from it. Therefore, the agent will choose a different action to avoid the fire pit.

We were also unable to achieve faster convergence on a test map with an agent pre-trained on fire maps, unlike the ladder maps. This could be due to fire pits—unlike ladders—actually kill the player causing it to start over again. Tweaking the hyper-parameters and settings should be able to give better results, i.e, faster convergence on a test map.

We hope that our work inspires further investigation into unravelling human world knowledge and applying it to new domains such as games.

References

Bengio, Yoshua et al. (2009). "Curriculum learning". In: *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48.

Dorigo, Marco and Marco Colombetti (1998). *Robot shaping: an experiment in behavior engineering*. MIT press.

Doshi-Velez, Finale and Zoubin Ghahramani (2011). "A comparison of human and agent reinforcement learning in partially observable domains". In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. Vol. 33. 33.

Dubey, Rachit et al. (2018). "Investigating human priors for playing video games". In: *arXiv preprint arXiv:1802.10217*.

Elman, Jeffrey L (1993). "Learning and development in neural networks: The importance of starting small". In: *Cognition* 48.1, pp. 71–99.

Florensa, Carlos et al. (2017). "Reverse curriculum generation for reinforcement learning". In: *arXiv preprint arXiv:1707.05300*.

Gureckis, Todd M and Bradley C Love (2009). "Learning in noise: Dynamic decision-making in a variable environment". In: *Journal of Mathematical Psychology* 53.3, pp. 180–193.

Justesen, Niels and Sebastian Risi (2018). "Automated curriculum learning by rewarding temporally rare events". In: *2018 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, pp. 1–8.

Kattan, Michael W, Dennis A Adams, and Michael S Parks (1993). "A comparison of machine learning with human judgment". In: *Journal of Management Information Systems* 9.4, pp. 37–57.

Kober, Jens, J Andrew Bagnell, and Jan Peters (2013). "Reinforcement learning in robotics: A survey". In: *The International Journal of Robotics Research* 32.11, pp. 1238–1274.

Kulkarni, Tejas D et al. (2016). "Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation". In: *Advances in neural information processing systems*, pp. 3675–3683.

Lake, Brenden M et al. (2017). "Building machines that learn and think like people". In: *Behavioral and brain sciences* 40.

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Alex Graves, et al. (2013). "Playing Atari with Deep Reinforcement Learning". In: *NIPS Deep Learning Workshop 2013*.

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei A Rusu, et al. (2015). "Human-level control through deep reinforcement learning". In: *Nature* 518.7540, pp. 529–533.

Muggleton, Stephen et al. (1989). "An experimental comparison of human and machine learning formalisms". In: *Proceedings of the sixth international workshop on Machine learning*. Elsevier, pp. 113–118.

Pathak, Deepak et al. (2017). "Curiosity-driven exploration by self-supervised prediction". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 16–17.

Peterson, Gail B (2004). "A day of great illumination: BF Skinner's discovery of shaping". In: *Journal of the experimental analysis of behavior* 82.3, pp. 317–328.

Portelas, Rémy et al. (2020). "Automatic Curriculum Learning For Deep RL: A Short Survey". In: *arXiv preprint arXiv:2003.04664*.

Rohde, Douglas LT and David C Plaut (1999). "Language acquisition in the absence of explicit negative evidence: How important is starting small?" In: *Cognition* 72.1, pp. 67–109.

Schmidhuber, Jürgen (2015). "Deep learning in neural networks: An overview". In: *Neural networks* 61, pp. 85–117.

Silver, David et al. (2016). "Mastering the game of Go with deep neural networks and tree search". In: *nature* 529.7587, p. 484.

Skinner, BF (1938). *The behavior of organisms: an experimental analysis. Appleton-Century*.

Sutton, Richard S and Andrew G Barto (2018). *Reinforcement learning: An introduction*. MIT press.

Svetlik, Maxwell et al. (2017). "Automatic curriculum graph generation for reinforcement learning agents". In: *Thirty-First AAAI Conference on Artificial Intelligence*.

Tan, Ming (1993). "Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents". In: *In Proceedings of the Tenth International Conference on Machine Learning*. Morgan Kaufmann, pp. 330–337.

Tesauro, Gerald (1995). "Temporal difference learning and TD-Gammon". In: *Communications of the ACM* 38.3, pp. 58–68.

Tsividis, Pedro A et al. (2017). "Human learning in Atari". In: *2017 AAAI Spring Symposium Series*.

Wellman, Henry M and Susan A Gelman (1992). "Cognitive development: Foundational theories of core domains". In: *Annual review of psychology* 43.1, pp. 337–375.

Wu, Yuxin and Yuandong Tian (2016). "Training agent for first-person shooter game with actor-critic curriculum learning". In: