Challenge Reinforcement Learning with Interactive Curriculum

Jiali Duan, Yilei Zeng, Yang Li
Emilio Ferrara, Lerrel Pinto, C.-C. Jay Kuo, Stefanos Nikolaidis
{jialidua,yilei.zeng,yli546,emiliofe,nikolaid}@usc.edu
lerrel@cs.nyu.edu, cckuo@sipi.usc.edu

Abstract

Curriculum reinforcement learning benefits training by focusing on examples of gradual increasing difficulty that are neither too hard nor too easy [1]. However, it's not always intuitive to define a curriculum that lies within this "range of interest". On one end of spectrum, a SOTA algorithm that learns from scratch may fail to collect any reinforcing signal. While on the other end, an automatic task-agnostic curriculum could "overfit" in an early phase that prevents it from further adaptation. In contrast, human has an innate ability to improvise and adapt when confronted with different scenarios, which we utilize to provide explainability and guidance for curriculum reinforcement learning. We first identify the "inertial" problem in automatic curriculum and then propose a simple interactive curriculum framework that works in environments that require millions of interactions. Demo and executable is available[2].

1 Introduction

A curriculum organizes examples in a more meaningful order which illustrates gradually more concepts, and more complex ones so that humans and animals can learn better [Bengio *et al.*, 2009]. When combined with reinforcement learning, it's been shown that a curriculum can improve convergence or performance compared to learning from the target task from scratch [Taylor and Stone, 2009; Graves *et al.*, 2017; Florensa *et al.*, 2017].

Previous works [Bengio *et al.*, 2009; Held *et al.*, 2018] showed that to reap the advantage of a curriculum strategy, the most beneficial examples are those that are neither "too hard" nor "too easy". One question that naturally arises then is how to design a metric to quantify how hard a task is so that we can sort tasks accordingly? A line of work deals with this problem by proposing curriculum automatically through another RL agent, such as teacher-student framework [Matiisen *et al.*, 2019; Portelas *et al.*, 2019], self-play [Sukhbaatar *et al.*, 2017; Bansal *et al.*, 2017; Baker *et al.*, 2019] or goal-gan [Held *et al.*, 2018]. One way of interpreting these approaches is that curriculum evolves through the adversarial nature between the two agents, similar to GAN [Goodfellow *et al.*, 2014; Duan *et al.*, 2018].

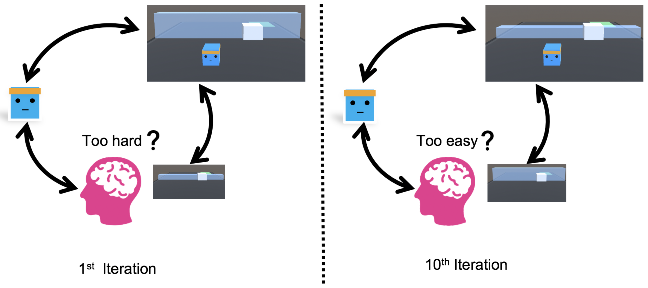


Figure 1: Given specific scenarios during curriculum training, humans can adaptively decide whether to be "friendly" or "adversarial" by observing the progress the agent is able to make. In cases where performance degrades, a user may flexibly adjust the strategy as opposed to an automatic assistive agent.

However, it's not always possible to formulate a curriculum through self-play (e.g., NPC in Wall-Jumper task) or require that the parameter space be continuous for adversarial learning (e.g, GridWorld task). In these cases, automatic curriculum most commonly used transitions from easy to hard. Secondly, it's possible that in certain cases the primary agent can learn better when another agent present is "friendly" rather than "adversarial", as in [Rusu *et al.*, 2016]. Compared to an automatic agent, human has an innate ability to improvise and adapt when confronted with different scenarios, which we utilize to provide expalinability and flexibility for curriculum reinforcement learning. In Figure 1, a user is able to intuitively understand the learning progress and dynamically manipulate the task difficulty by changing the height of the wall. With new challenging environments, we show how human inductive bias can help solve three nontrivial tasks that are otherwise unsolvable by learning from scratch or even auto-curriculum.

In Section 2, we give a brief introduction of related work. In Section 3, Our interactive curriculum platform is intro-

[1] [Bengio *et al.*, 2009] pointed out with empirical experiments that examples which are "too easy" do not help much to improve the model, whereas examples considered "too difficult" would not be captured by small change in the model.

[2] https://github.com/davidsonic/interactive-curriculum-reinforcement-learning

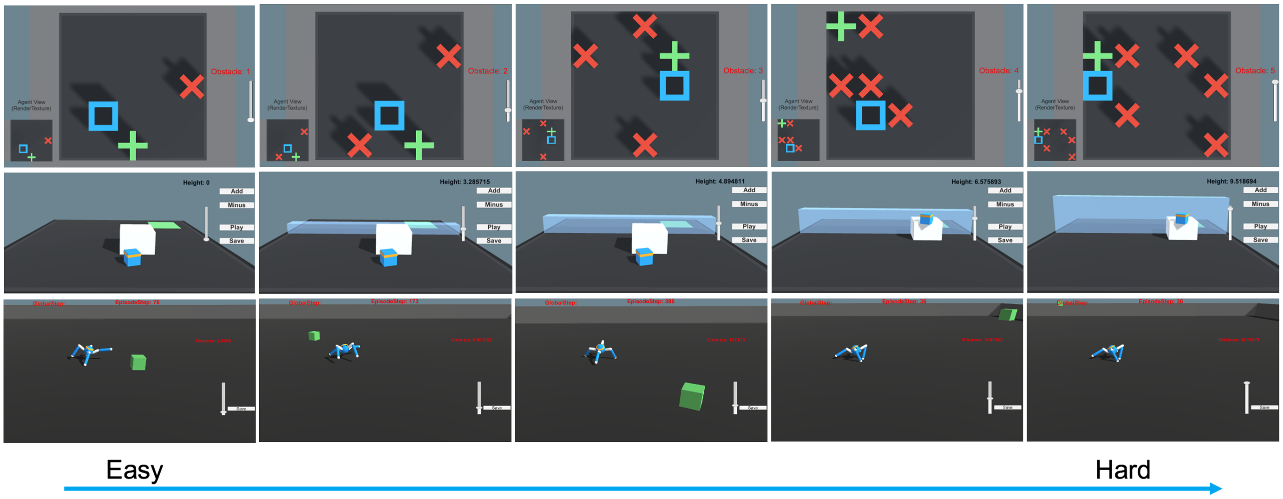


Figure 2: Our interactive platform for curriculum reinforcement learning, in which the user is allowed to manipulate the task difficulty via a unified interface (slider and buttons). All three tasks receive only sparse rewards. The manipulable variable for the three environments are respectively the number of red obstacles (GridWorld, Top row), the height of wall (Wall-Jumper, Middle row) and the radius of the target (SparseCrawler, Bottom row). The task difficulty gradually increases from left to right.

duced, with which we identify the "inertial" problem in an "easy-to-hard" automatic curriculum. In Section 4, We show preliminary results of user studies on our environments that require millions of interactions. Finally, we conclude and discuss future work in Section 5.

2 Related Work

2.1 Curriculum Reinforcement Learning

[Florensa *et al.*, 2017] learns to reach goals with sparse rewards by following reverse curriculum, generated by start states that grow increasingly far from the goal. [Bansal *et al.*, 2017; Baker *et al.*, 2019] explores adversarial self-play to automatically evolve curriculum in playing sumo, football etc. [Matiisen *et al.*, 2019; Portelas *et al.*, 2019] formalizes teacher-student transfer learning framework, where a student agent works on actual tasks while a teacher network is tasked with selecting tasks. The most related work to ours is [Heess *et al.*, 2017], which shows empirically how a rich environment can help to promote the learning of complex behavior without explicit reward guidance. In comparison, we evolve environments leveraging human's inductive bias in curriculum design.

2.2 Human-in-the-Loop Reinforcement Learning

As learning agents move from research labs to the real world, it becomes increasingly important for human users especially those without programming skills, to teach agents desired behavior. A large amount of work focuses on imitation learning [Schaal, 1999; Ross *et al.*, 2011; Ho and Ermon, 2016; Pinto and Gupta, 2016], where demonstrations from the expert act as direct supervision. Humans can also interactively shape training with only positive or negative reward signals [Knox and Stone, 2009] or combine manual feedback with rewards from MDP [Knox and Stone, 2010; Abel *et al.*, 2017]. A recent work formulates human-robot interaction as an adversarial game [Duan *et al.*, 2019] and shows improvement of grasping sucess and robustness when the robot trains with a human adversary.

In this paper, we aim to close the loop between these two fields, by studying the effect of interactive curriculum on reinforcement learning. To achieve this, we have designed three challenging environments that are nontrivial to solve even for state-of-the-art RL method [Schulman *et al.*, 2017], which we describe in the next Section.

3 Interactive Curriculum Guided by Human

3.1 Interactive Platform

Figure 2 shows our released environments for curriculum reinforcement learning, where the task difficulty can be manipulated by users. The agents are expected to reach the green target in GridWorld, navigate to land on green mat in Wall-Jumper and reach dynamic green box in SparseCrawler respectively. Our interactive platform is built with three goals in mind: 1) Real-time online interaction with flexibility; 2) Parallelizable for human-in-the-loop training; 3) Seamless control between reinforcement learning and human-guided curriculum.

To achieve the first goal, an event-driven environment container is run separated from the training process, allowing user to send a control signal (e.g., UI control, scene layout, task difficulty) to the environment at any time during training via interactive interface. To achieve similar efficiency as automatic training, we integrate human-interactive signal into RL parallelization. On the one hand, centeralized SGD update with decentralized experience collection is performed as agents of the same kind share the same network policy [Mnih *et al.*, 2016]. On the other hand, we enable controlling environment parameters in different instantiations simultaneously via a unified interactive interface, which makes it possible to solve tasks that require millions of interactions. For the third goal, we display real-time instructions and allow users to inspect learning progress when designing curriculum.

3.2 A Simple Interactive Curriculum Framework

Curriculum reinforcement learning is an adaptation strategy to improve RL training by ordering a set of related tasks to be learned [Bengio *et al.*, 2009]. The most natural ordering is to gradually increase the task difficulty with an automatic curriculum. However as shown in Figure 3a, the auto-curriculum quickly mastered skills when walls are low but failed to adapt when a dramatic change of skill is required (Figure 3c), leading to a degradation of performance on the ultimate task (Figure 3b). The reason is that the agent must use a box to navigate a high wall in contrast to low-wall scenarios, where additional steps to locate the box will be penalized.

Algorithm 1: Human-Guided Interactive Curriculum

```
Result: Agent's policy $\pi^R$
Initialize difficulty=0;
while $step \leq total\_step$ do
    $\pi^R_{new}$ = Train($\pi^R_{old}$, difficulty);
    if $step \ \% \ interval == 0$ then
        difficulty=$\mathcal{H}$ ($\pi^R_{new}$, difficulty);
    end
    $\pi^R{}_{old} = \pi^R{}_{new}$
end
```

Our results testify what [Bengio *et al.*, 2009] observed in their curriculum for supervised classification task, that curriculum should be designed to focus on “interesting” examples. In our case, curriculum that resided at an easy level for the first 3M steps “overfitted” to the previous skill and prevented it from adapting. Although a comprehensive IF-ELSE rule is possible, in real-world where situations could be arbitrarily complex, adaptable behavior out of guidance from human is desired. Following this spirit, we test the ability of human interactive curriculum using a simple framework (Algo 1), where human (function $\mathcal{H}$) provides feedback by adjusting the task difficulty at fixed interval in the training loop (i.e., after evaluating the agent's learning progress on current difficulty, user can choose to tune the task easier/harder or leave it unchanged). We show in the next Section that with this simple interactive curriculum, tasks that are originally unsolvable can be guided towards success by human, with an additional property of better generalization.

4 Experiments

We train the agents for three competitive tasks using the training method described previously. Our aim is to show that human-in-the-loop interactive curriculum are capable of leveraging human prior during adaptation which allows agents to build on past experiences. For all our experiments, we fix the interaction interval (e.g, 0, 0.1, 0.2,...,0.9 of the total steps) and allow users to inspect learning progress twice before adjusting the curriculum. The user can either choose to make it easier, harder or unchanged. Our baseline is PPO with the optimized parameters as in [Juliani *et al.*, 2018]. We train GridWorld, Wall-Jumper and SparseCrawler for 50K, 5M and 10M steps respectively.

4.1 Effect of Interactive Curriculum

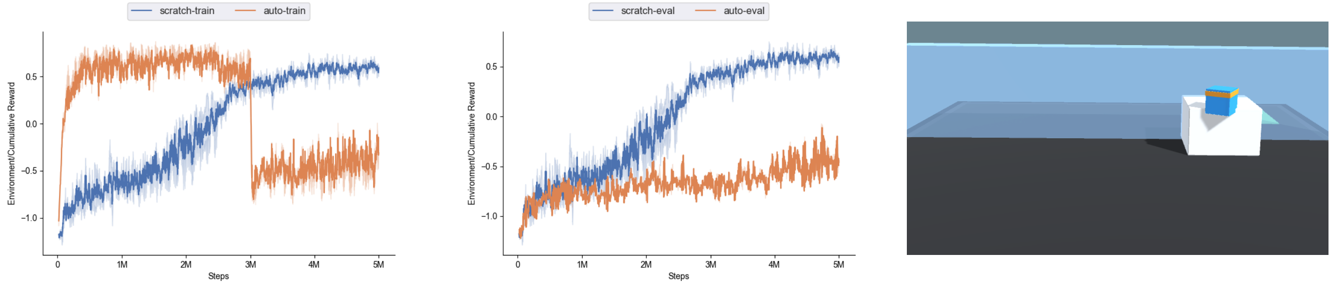
In Section 3.1, we introduced three challenging tasks due to sparsity of rewards. For example in Figure 4a, we observed that agents which learn from scratch (green and red curves) had little chance of success with obstacles scattered around the grid, thus failing to reinforce any desired behavior. On the other hand, users were able to gradually load or remove obstacles by inspecting the learning progress. Eventually, the models trained with our framework are able to solve GridWorld with 5 obstacles present. Inspired by this, we further tested our framework on SparseCrawler task (Figure 4c), which requires 10M steps of training. Thanks to our parallel design (Section 3.1), we were able to reduce the training time from 10 to 3 hours during which users would interact 10 times. When trained with dynamically moved targets of increasing radius, we found that crawlers gradually learned to align themselves toward the right direction.

In the Wall-Jumper task (Figure 4b), we noticed a variance of performance given different users. One run (blue curve) outperformed learning from scratch with an obvious margin while another run (orange curves) performed less well but still converged as learning from scratch. Nevertheless, both of the two trials are much better than an auto-curriculum that suffers from over-fitting as described in Section 3.2.

4.2 Generalization Ability

Over-fitting to a particular dataset is a common problem in supervised learning. Similar problems can occur in reinforcement learning when there's no or little variation in the environment. To deal with this problem, we had considered: 1) randomness in terms of how grid is generated; layout of blocks and jumpers; locations of crawlers and targets. 2) entropy regularization in our PPO implementation, making a strong baseline.

We compare models trained with our framework with ones trained from scratch in three environments with a set of tasks. For example, in GridWorld the agents were tested with the number of obstacles increasing from 1 to 5. In Wall-Jumper, the heights of the wall rise from 0 to 8 discretely during testing and in SparseCrawler, the radius of the moving target transitions from 5 to 40 with a span of 5 (Figure 5). One common observation is that our model consistently outperforms learning from scratch. Secondly, there's a large gap between the curves from curriculum model and learning from scratch (Figure 5a), indicating that they “warm-up” more quickly with easy tasks than directly jumping into the difficult task. This is analogous to how human learns by building on past experiences. Interestingly, the curves eventually congregate in Wall-Jumper (Figure 5b), for both curriculum model and scratch model. Finally, we observed that the performance of our model in SparseCrawler (Figure 5c) continually arose and reached the target with 1 to 2 more success, as opposed to Wall-Jumper environment. The reason is because we would reset the environment in SparseCrawler only when it reaches the maximum time steps in a single round.

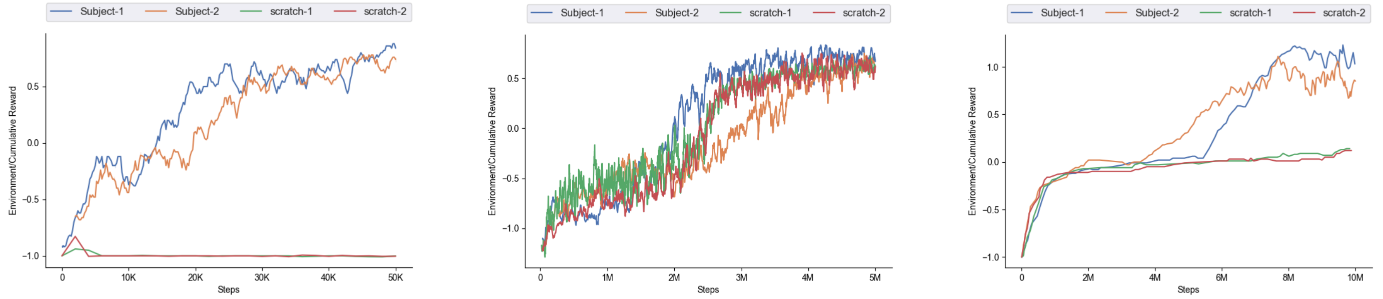


(a) Training curve

(b) Testing curve

(c) High Wall

Figure 3: "Inertial" problem of auto-curriculum which gradually grows the difficulty at fixed interval. The performance of auto-curriculum (orange curve) drops significantly when navigation requires jumping over the box first but the learning inertial prevents it from adapting to the new task. Note that testing curve is evaluated on the ultimate task unless otherwise stated.

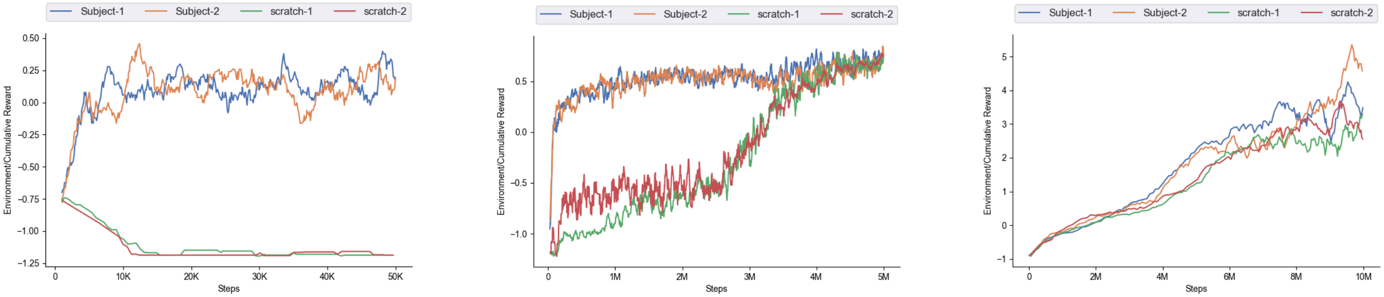


(a) GridWorld (obstacles of 5)

(b) Wall-Jumper (height of 8)

(c) SparseCrawler (radius of 40)

Figure 4: **Effect of interactive curriculum evaluated on the ultimate task.**



(a) GridWorld (obstacles from 1 to 5)

(b) Wall-Jumper (heights from 0 to 8)

(c) SparseCrawler (radius from 5 to 40)

Figure 5: **Generalization ability of interactive curriculum evaluated on a set of tasks. The average performance over these tasks is plotted for different time steps.**

When performing qualitative tests, our model solves the GridWorld with varying obstacles whereas scratch model fails when the number of obstacles exceeds 3. For Wall-Jumper, our model is able to reach the goal with minimum steps while the scratch model would inevitably use the block, necessary only for heights over 6.5. In the SparseCrawler environment, our model has a faster moving speed and a greater numbers of success whereas the scratch model could only reach proximal targets.

5 Conclusion

In this paper, we released three new environments that are challenging to solve (sparse reward, transfer between skills and large amount of training up to 10M steps), with varying curriculum space (discrete/continuous). With this environment, we identified a phenomenon of over-fitting in auto-curriculum that leads to deteriorating performance during skill transfer. Then, We proposed a simple interactive curriculum framework facilitated by our unified user interface. Experiment shows the promise of a more explainable and generalizable curriculum transition by involving human-in-the-loop, on tasks that are otherwise nontrivial to solve. For future work, we would like to explore the effect of human function on the final performance and to provide more source as reference for users' decision-making.

References

[Abel *et al.*, 2017] David Abel, John Salvatier, Andreas Stuhlmüller, and Owain Evans. Agent-agnostic human-in-the-loop reinforcement learning. *arXiv preprint arXiv:1701.04079*, 2017.

[Baker *et al.*, 2019] Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula. *arXiv preprint arXiv:1909.07528*, 2019.

[Bansal *et al.*, 2017] Trapit Bansal, Jakub Pachocki, Szymon Sidor, Ilya Sutskever, and Igor Mordatch. Emergent complexity via multi-agent competition. *arXiv preprint arXiv:1710.03748*, 2017.

[Bengio *et al.*, 2009] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

[Duan *et al.*, 2018] Jiali Duan, Xiaoyuan Guo, Yuhang Song, Chao Yang, and C-C Jay Kuo. Portraitgan for flexible portrait manipulation. *arXiv preprint arXiv:1807.01826*, 2018.

[Duan *et al.*, 2019] Jiali Duan, Qian Wang, Lerrel Pinto, C-C Jay Kuo, and Stefanos Nikolaidis. Robot learning via human adversarial games. *CoRR*, 2019.

[Florensa *et al.*, 2017] Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. Reverse curriculum generation for reinforcement learning. *arXiv preprint arXiv:1707.05300*, 2017.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[Graves *et al.*, 2017] Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1311–1320. JMLR. org, 2017.

[Heess *et al.*, 2017] Nicolas Heess, Dhruva TB, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, SM Eslami, et al. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*, 2017.

[Held *et al.*, 2018] David Held, Xinyang Geng, Carlos Florensa, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents. 2018.

[Ho and Ermon, 2016] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in neural information processing systems*, pages 4565–4573, 2016.

[Juliani *et al.*, 2018] Arthur Juliani, Vincent-Pierre Berges, Esh Vckay, Yuan Gao, Hunter Henry, Marwan Mattar, and Danny Lange. Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627*, 2018.

[Knox and Stone, 2009] W Bradley Knox and Peter Stone. Interactively shaping agents via human reinforcement: The tamer framework. In *Proceedings of the fifth international conference on Knowledge capture*, pages 9–16, 2009.

[Knox and Stone, 2010] W Bradley Knox and Peter Stone. Combining manual feedback with subsequent mdp reward signals for reinforcement learning. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 5–12. Citeseer, 2010.

[Matiisen *et al.*, 2019] Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher-student curriculum learning. *IEEE transactions on neural networks and learning systems*, 2019.

[Mnih *et al.*, 2016] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

[Pinto and Gupta, 2016] Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 3406–3413. IEEE, 2016.

[Portelas *et al.*, 2019] Rémy Portelas, Cédric Colas, Katja Hofmann, and Pierre-Yves Oudeyer. Teacher algorithms for curriculum learning of deep rl in continuously parameterized environments. *arXiv preprint arXiv:1910.07224*, 2019.

[Ross *et al.*, 2011] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635, 2011.

[Rusu *et al.*, 2016] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[Schaal, 1999] Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6):233–242, 1999.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[Sukhbaatar *et al.*, 2017] Sainbayar Sukhbaatar, Zeming Lin, Ilya Kostrikov, Gabriel Synnaeve, Arthur Szlam, and Rob Fergus. Intrinsic motivation and automatic curricula via asymmetric self-play. *arXiv preprint arXiv:1703.05407*, 2017.

[Taylor and Stone, 2009] Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(Jul):1633–1685, 2009.