Exploring Historical Self Play for Autocurricula Generation

Reinforcement learning (RL) details a framework for agents to interact with the environment in order to optimize rewards. Agents often encounter tasks where information is limited. Therefore, developing sample-efficient means of exploring an environment is important, especially for those that reward-sparse. By using a form of adversarial learning called Asymmetric Self-Play, it is possible to frame the task of curriculum generation as an RL objective which can guide a student agent to better explore an environment. This two-player game between RL agents can be unstable. We propose using historical averaging, a game-theoretic result from the study of fictitious play that induces convergence to Nash Equilibria, with self-play to introduce a new method: 'Historical Self Play.'

Historical Self Play

An adversarial method of autocurricula generation, called asymmetric self-play, constructs a new teaching agent that itself learns under an RL algorithm. We consider two agents, a student and a teacher, where the latter proposes a task to complete to the former. This task is chosen by the teacher to encourage learning by progressively providing more difficult tasks that are beyond the current capabilities of the student agent but not too difficult. This effectively is a learning curriculum that can teach an agent how to traverse the environment. We can induce this kind of learning by constructing certain reward functions for the teacher:

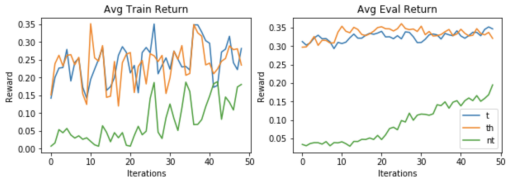
$$\tilde{R}_{\text{teacher}} = \gamma \max\{0, R_{\text{teacher}} - E_{s\sim\pi_{\text{student}}}(R_{\text{student}})\}$$

Self-play itself is a strong tool to encourage learning; however, we have no guarantees of convergence to an equilibrium strategy. We proceed to refine the application of self-play by considering *fictitious play* and *historical averaging* to improve the stability of this adversarial learning setup. By formulating an augmented adversarial learning scheme called 'Historical Self Play', we explore various historical averaging setups of Self-Play and consider their general effectiveness compared to traditional methods, thus providing a notion of the practicality of these methods.

Experiments and Results

To perform our experiments, we use a grid-world setup to analyse how these methods perform on discrete settings. By using an intuitive approach for initial experiments for choosing how to undergo historical, specifically by using an exponential sampling distribution to favour more recent weights and used inter-trajectory historical averaging to play more accurately against past policies, we found similar performance between agents that have a historical self-play scheme and those with a regular self-play scheme. With the additional convergence guarantees associated with historical averaging, we see that this often forgotten step in adversarial training is potentially worthwhile.

Figure 1: Comparisons of Historical Self-play Performance against Traditional Self-Play Methods



The above figure shows comparisons of the performance between regular RL learning (nt), teaching (t), and teaching with historical averaging (th).

Discussion

This work provides a proof of concept exploring how following the theory associated with adversarial settings, specifically fictitious play, can provide similar performance results to traditional adversarial methods like self-play. While considering an appropriate model for historical averaging, historical self-play can maintain stronger convergence guarantees. This shows that in more difficult settings, such as environments with continuous state-action spaces, we can leverage similar techniques to have the same assurances for adversarial training.

Exploring Historical Self Play for Autocurricula Generation

Anand Siththaranjan
University of California, Berkeley
anandsranjan@berkeley.edu *

Sandy Tanwisuth
University of California, Berkeley
kst@berkeley.edu *

Abstract

Sample-efficient means of exploration are important for an agent to effectively complete their tasks in large, sparse reward environments. By using a form of adversarial learning previously called Self-Play, the task of curriculum generation is framed as a reinforcement learning (RL) objective which can guide a student agent to better explore an environment and results in better performance. As this is a two-player game between RL agents, this type of learning can be unstable. We propose using historical averaging, a game-theoretic result from study of fictitious play that induces convergence to *Nash Equilibria*, as a solution. We introduced an augmented adversarial learning scheme called 'Historical Self Play' (HSP), and explore various historical averaging setups of Self-Play to consider their general effectiveness and practicality compared to traditional methods. Our results showed that HSP performs similarly to current adversarial methods and better than regular RL techniques, showing that historical averaging techniques can potentially be leveraged to provide convergence assurances for adversarial training without degrading performances. An implementation can be found on our website [2].

1 Introduction

Reinforcement learning (RL) details a framework for agents to interact with the environment to optimize rewards. In RL, agents often encounter tasks where information is limited and thus little information can be inferred from the environment. Key examples of this are sparse reward settings where rewards are highly concentrated and sparse, inducing a reward landscape that makes it difficult for an agent to learn. As a result, it is often vital for the underlying algorithms of these agents to encourage exploration of these environments so as to be able to find these high reward states.

To deal with this, there are many methods that have been developed which encourage explorations [Fu et al. [2017]; Mnih et al. [2015]; Lillicrap et al. [2016]]. However, we can consider an alternative approach: one that generates a curriculum [Bengio et al. [2009]; Leibo et al. [2019]] from which the 'student' agent can learn. Rather than blindly encourage just exploration of an environment, a curriculum is more effective at teaching a learner skills that may be transferable under various settings. This is often seen in supervised learning settings, where for example in multiple regression tasks for supervised learning rather than randomly provide batches of data as in stochastic gradient descent, we use pre-selected batches that can progressively help a learner understand various aspects of the space to be regressed on. Similarly, this can also be applied to RL contexts. By training an agent on tasks of increasing complexity and in a certain order, they can more effectively ascertain the reward landscape and could be more effective in approaching a variety of tasks.

*Anand and Sandy devised the project and provided each other directions through all stages of the work. Anand derived the implementation, and obtained the first results. Sandy retested the experiments to confirm the results and provided feedback for improvements. We thank Logan Cross and Kevin Xu for their valuable discussions.

[2]Project website :https://github.com/AnandS29/AsymmPlay

However, the question of how to effectively develop these curriculum is difficult. For RL, one solution is to allow for intervention from an expert, who would need to analyse the environment in question and choose an appropriate series of tasks that, to this expert, makes the most sense. In supervised learning, it has been shown [Bengio et al., 2009] that gradually introducing more difficult examples, can speed up the online training or improve test accuracy. Though there are cases where it is simple to determine how 'difficult' an example is, such as if the level of noisiness of the data were known, other examples such as geometry or images can sometimes need human discretion to choose. In settings, where the space of possibilities is large, such as environments with large state-action spaces, this is difficult to do. As such, autocurricula generation aims to develop these curriculum mechanically to become a technique that is generalizable to variety of scenarios.

The focus of this paper is to analyse an adversarial method of autocurricula generation, called self-play [Sukhbaatar et al., 2017]. Specific to reinforcement learning, this method constructs a new agent that itself learns under an RL algorithm. The goal of this agent, which we call the teaching agent, is to construct tasks that a student agent will learn from by training specifically on that task. As the student's ability to undertake a task improves, the teacher is incentivised via a specifically-designed reward function to give progressively more difficult, but not too difficult, tasks. This provides a learning curriculum that can teach an agent how to traverse the environment.

As this is an adversarial method, we have two-player game consisting of a teacher and student player. As such, we can consider a notion of equilibrium as traditionally defined in game theory to be a point at which neither player has anything to gain by changing their current strategies. From the perspective of reinforcement learning, this can be seen as a point in the learning process where an agent's policy converges and thus ceases to update.

Due to this being an adversarial approach to generating curriculum, training of agents with multi-layer perceptron policies can be difficult to do effectively. This has been seen previously with generative adversarial networks (GANs) [Goodfellow et al., 2014], which are difficult to train due to the need to optimize two networks that need to interact with each other. As a result, issues such as oscillating model parameters, diminishing gradient, etc. are common to see. Similarly, difficulties can be experienced when training two competing agents in an RL setting. To help stabilize play, we propose using *historical averaging* [Lee et al., 2019] as a means by which to stabilize learning for both agents. This concept comes out of fictitious play, an idea stemming from game theory which suggests that when both agents utilize their past policies when making decisions, they can have convergence guarantees to *Nash equilibria*. This added step to the adversarial learning process is often forgone, and in this work we explore the efficacy of *historical self-play* with regards to traditional self-play under various setups.

2 Related Works

2.1 Adversarial Learning

[Goodfellow et al., 2014] purposed Generative Adversarial Networks (GANs) methods for deep learning by which utilizing competition from different networks without annotated training data. Similarly to GANs, our approach contains two networks equipped with different tasks . The networks in our approach are the teacher and the student networks, which are similar to generators and discriminators for GANs, where the teacher has the goal of generating tasks while the student has the goal of completing the tasks created by the teacher rather than discriminates. The construction of both are similar but differ in their application, with the former's goal is to find a generator while the latter's is to effectively train the student.

2.2 Curriculum Learning

Curriculum training is method of training where agents can learn from generated sequences of obstacles [Bengio et al., 2009].) Previous methods require manual labeling; however, recent developments [Florensa et al. [2017a]; Florensa et al. [2017b]; Leibo et al. [2019]] have introduced self-generated auto-curriculum. Similarly to recent approaches, our method is unsupervised as the teacher is the one generating the curriculum. We differ by considering an adversarial game between RL agents as our method for curriculum generation.

2.3 Self-Play

Self-play has long been applied in RL contexts [Samuel [1959]; Tesauro [1995]; Silver et al. [2016]]. Unlike previous approaches, [Sukhbaatar et al., 2017] has introduced an asymmetric approach in self-play where by the networks are divided into the task-setter and the normal learning agent. The task-setter is equivalent to our teacher, whereas the agent is equivalent to our student. We consider a similar approach that uses expected rewards of the student that the teacher optimizes against, as well as a simpler game whereby the student does not need to undo every task but just has to reach goals.

2.4 Fictitious Play

Fictitious play [Robinson, 1951] is a classic procedure from game theory where the players choose best responses to their opponent's average behavior. This method has been adapted to deep reinforcement learning where Full-Width Extensive-Form Fictitious Play (XFP) [Heinrich et al. [2015]] was introduced to enable fictitious players to update their strategies such that linear time and space complexity is accomplished. Extension work in Fictitious Self-Play [Heinrich and Silver, 2016] is a class of algorithms that approximate XFP. Unlike these works, we applied the Fictitious Play in a discrete time rather than in a continuous time as shown in these works. Our approach to implementing this is similar to that suggested in [Lee et al., 2019], which simply keeps track of historical network policies and appropriately samples.

3 Asymmetric Self-Play

In asymmetric self-play we consider two agents: a student and a teacher. Both agents have separate minds, such that they have different policies, π_{student} and π_{teacher} respectively, and goals to optimize. During an iteration of teaching, the goal of the student agent is to learn to accomplish some task that is proposed by the teacher agent. This task is chosen by the teacher to encourage learning by progressively providing more difficult tasks that are beyond the current capabilities of the student agent. However the teacher should not want to provide too difficult of a task, and we can induce this kind of learning by constructing a certain reward function for the teacher:

$$\tilde{R}_{\text{teacher}} = \gamma \max\{0, R_{\text{teacher}} - E_{s\sim\pi_{\text{student}}}(R_{\text{student}})\}$$

We note that the standard reward R_{agent} is the reward for an agent for completing the task proposed. Then the teacher's reward is set to be proportional to the $\max\{0, R_{\text{teacher}} - E_{\pi_{\text{student}}}(R_{\text{student}})\}$, which tries to encourage the teacher to create a difficult task that adversarially optimizes against the student's average reward. The $\max\{0, R_{\text{teacher}} - E_{\pi_{\text{student}}}(R_{\text{student}})\}$ makes it such that the task is not too difficult, while the γ allows us to scale the reward the teacher receives depending on the environment. This reward mechanism is similar to that seen in [Sukhbaatar et al., 2017], but we choose to use an expected reward over the trajectory distribution induced by the policy so as to train more efficiently as an adversary by being less susceptible to randomness in the student's policy. This can be estimated in practice by sampling trajectories and taking averages of rewards for the student trained on the proposed task.

For the teacher to be able to propose tasks to the student, we must consider environments where they are either (a) resettable or (b) reversible. A resettable environment is one that allows the teacher to undertake their proposed task, and reset the environment to its original state such that the student can thus undertake the same task in the same initial conditions. For a reversible environment, it allows for the student to undo any of the actions that changed the state of the environment. This is for reversible tasks where students learn to exactly undo the actions of the teacher and return to the same initial state of the teacher.

In this work, we consider tasks in a resettable environment where a teacher only proposes an end state or goal that must be achieved by the student from the same initial conditions. This is different from [Sukhbaatar et al., 2017], which mainly considers reversible tasks that can consist of many sub tasks that need to be "undone". As such, our tasks are simpler and allow for more flexibility in accomplishing the task and ease of training for the student. This could also allow for more innovative methods from the student to solve a task, or could force a student to inherently learn intermediate goals.

We proceed to refine the application of self-play by considering fictitious play and historical averaging to improve the stability of this adversarial learning setup, as well as goal-condition learning, which takes advantage of the task-based game play between agents.

3.1 Fictitious Play and Historical Averaging

Due to the adversarial nature of the learning, we consider game-theoretic concepts such as fictitious play to help in providing stronger convergence guarantees for our constructed two-player game and possibly produce more stable learning.

In the setting of discrete time fictitious play (DTFP) with two players, we can consider agents a_1 and a_2 that select a best response to each other's marginal distribution of strategies up to that point in time. This is what is known as fictitious play, as the agents construct a strategy that "plays" against the past empirical distributions of their opponents play. The following theory follows from analysis undertaken by Brown [Berger, 2007] and Robinson [Robinson, 1951], where the response associated to "fictitious play" is given by a distribution that weights all previous strategies of the agent. Though this analysis assumes that undertaking actions by the agents occur simultaneously, it has been noted in literature that there aren't significant qualitative differences between the assumption of alternating and simultaneous moves. By using this type of play, we can be assured that if there is convergence in strategy then the game will have converged to a Nash equilibrium.

We note that this idea of fictitious play must be undertaken by all agents participating in the game, and we detail how we can apply these concepts to reinforcement learning agents.

3.1.1 Application to Reinforcement Learning

We can apply these concepts to the learning of reinforcement learning agents by considering the historical weights of the model as the past strategies of the player. By keeping track of the historical models, we can select from these past models to be our current model and update the weights of it using a chosen RL algorithm. Together with self-play, we can establish HSP as an augmented method.

In practice, we choose a "probability of historical sampling" that limits the likelihood of sampling from past models. There are also two other parameters of choosing from the historical models that an agent needs to consider: the sampling distribution and when to sample a past model (either inter or intra trajectory).

3.1.2 Sampling Distributions

To undergo historical averaging under historical policies $\mathcal{P} = \{P_i\}_{i=1}^n$, we consider two probability distributions which determines how we sample: uniform and exponentially-weighted distributions.

Definition 3.1. Discrete Uniform Distribution

We define a discrete uniform distribution as one with probability mass function defined by:

$$\Pr(P_i) = \begin{cases} \frac{1}{n} & P_i \in \mathcal{P} \\ 0 & P_i \notin \mathcal{P} \end{cases}$$

Definition 3.2. Exponentially Weighted Distribution

We define an exponentially weighted distribution as one with probability mass function defined by:

$$\Pr(P_i) = \begin{cases} \frac{\alpha^i}{\sum_{j=1}^n \alpha^j} & P_i \in \mathcal{P} \\ 0 & P_i \notin \mathcal{P} \end{cases}$$

In practice, α is a number greater than 1 so as to weigh more recent policies greater than older policies.

This latter sampling method more favourably weighs more recent policies than early ones, which can allow for better performance than when considering all models equality, even those with little training, as is the case with the former method.

3.1.3 Inter-trajectory

In inter-trajectory historical averaging, we choose to sample from past models with some probability, and then proceed to use this model to roll out trajectories and subsequently update this model as per our learning algorithm.

3.1.4 Intra-trajectory

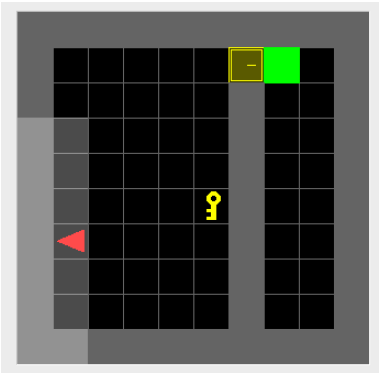
In intra-trajectory historical averaging, at the start of each step of our trajectories we sample from our past models, again with some probability. This is different from the inter-trajectory model of historical averaging as many policies can be used to roll out a trajectory rather than just a single one, as is the case for the previous method.

4 Experiments

4.1 Setup

We use a grid-world setup to analyse how these methods perform on discrete settings, an example of which is displayed in Figure 1. This grid-world is specifically an environment that allows the agent to access other parts of the space by picking up a key and subsequently unlocking the door. This adds a level of complexity to the task beyond just general exploration, and is similar to the environment used by [Sukhbaatar et al., 2017]. The agent also has a partially observable view of its states, as shown by the light grey "field of vision" in Figure 1.

Figure 1: Grid-world environment



An agent for teaching and historical averaging tasks learns by first pre-training with the teacher agent, who proposed goal states to reach, and then undergoes traditional training on a fixed goal. The non-teaching experiments only undergo the last task. Similar to [Sukhbaatar et al., 2017], we consider the pre-training tasks as "free", which they note to be commonly done in semi-supervised learning settings.

We consider various evaluation tasks to test out methods, specifically to compare how well the agent explores. We use three in particular: evaluation on the same goal as that in training, a different hand-picked goal, and many random goals.

We use different RL algorithms for each agent, based on our experimental findings. In particular we use Asynchronous Actor Critic algorithm (A2C) [Mnih et al., 2016] for the teacher agent and Proximal Policy Optimization algorithm (PPO) [Schulman et al., 2017] for the student. For the exponential distribution, we mimic the choice of variables as in [Lee et al., 2019], and thus use the value 1.2 as the value for the exponentiated variable α. The results shown are averaged over multiple random seeds.

4.2 Comparing Traditional RL Self-Play and Historical Self-Play

We show comparisons of the performance between regular RL learning (nt), teaching (t), and teaching with historical averaging (th), which is detailed in Figures 2-4. Our experiment for just teaching used

10 teaching iterations, each with 10 student iterations. For historical averaging we additionally use an exponential sampling distribution with probability of sampling 0.2, and undergo inter-trajectory historical averaging. For all experiments we use 50 non-teaching iterations.

Figure 2: Comparisons for evaluation on the same goal

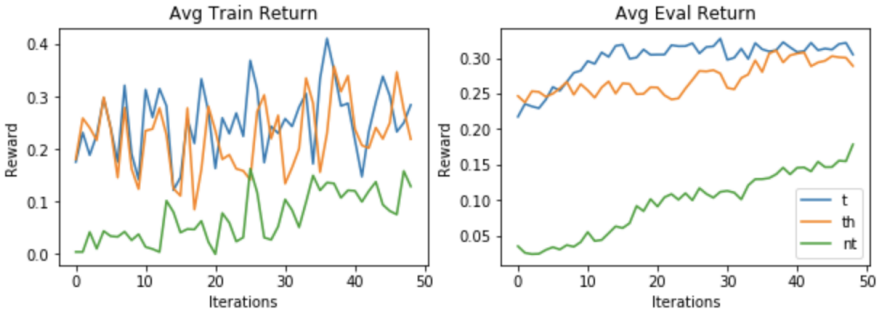


Figure 3: Comparisons for evaluation on the different goal

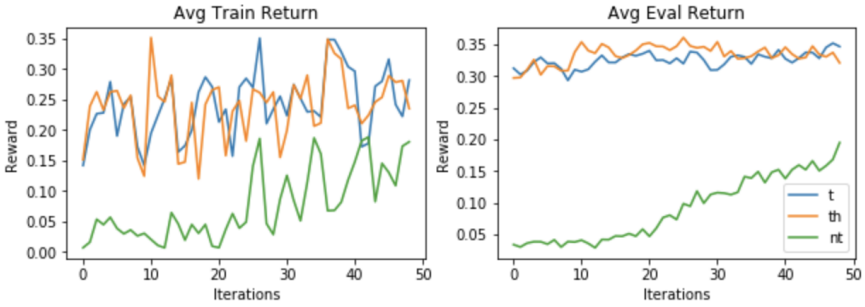
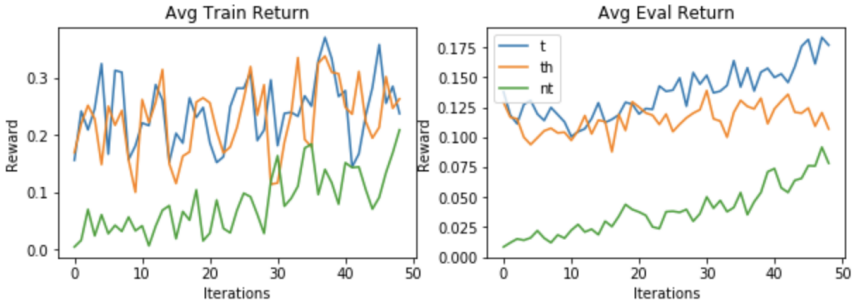


Figure 4: Comparisons for evaluation on multiple random goals



We note that pre-training with teaching improves the agents ability to undergo the various evaluation tasks and allows for training to begin with better performance. We see that there is little improvement for training, which could be due to the student agent learning an equilibrium strategy that is hard to change during the normal non-teaching phase. We also see that there is not much difference in training between using teaching and teaching with historical averaging, however when considering randomized evaluation tasks there is some divergence in performance. This could be explained by the fact that just teaching allows for greater exploration of the space, however when including historical averaging this is does not occur to the same degree as the agent learns against historical policies, which may not be optimal for giving exploratory tasks.

In the following results, we mainly see that performance is better on the random and different goals. As the initial training goal is difficult, we see this as a sign that the agent tries to learn the new goal

while being able to still find other easier goals. For many of the results, there is little change in training and evaluation, which suggest a convergence to an equilibrium strategy.

4.3 Historical Averaging

To analyse the various options for historical averaging, we compare using different sampling distributions (exponential and uniform), probabilities of sampling (0.05 and 0.4) and inter/intra historical averaging.

Sampling Distributions

Figures 5 and 6 detail the results of sampling with different distributions. We see that exponential performs better than uniform, which is to be expected due to it favouring more recently trained policies. As exponential also has stagnant evaluation curve while uniform is still learning, the former has likely converged to an equilibrium policy while the latter has not. This shows that an exponential distribution may be better suited to reaching an equilibrium strategy.

Figure 5: Historical averaging with an exponential sampling distribution

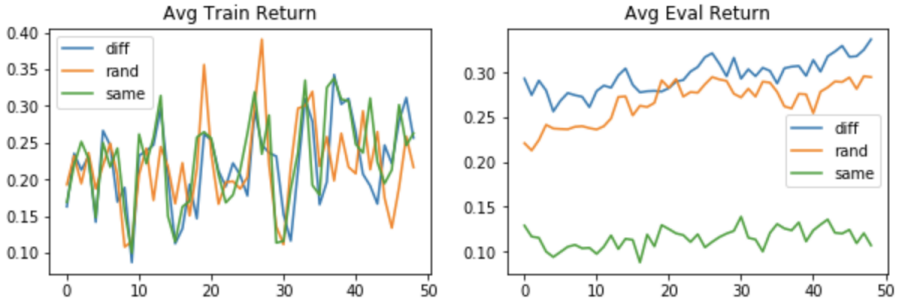
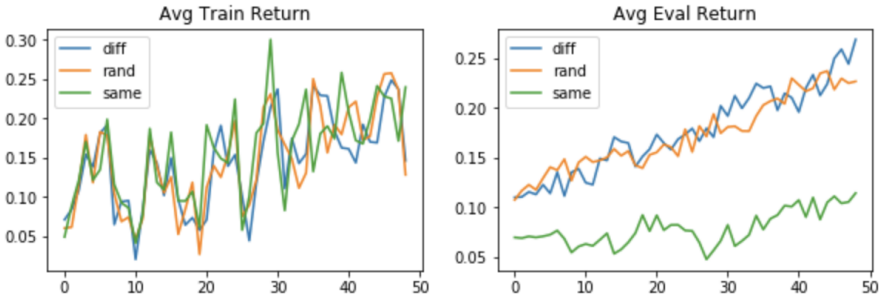


Figure 6: Historical averaging with a uniform sampling distribution



Probability of Sampling

Figures 7 and 8 show how results vary under various probabilities for sampling, using an exponential distribution as our sampling distribution. We note that this would intuitively mean that more frequent sampling would encourage earlier policies to be sampled more often, so lower probabilities would result in less variation in learning. We see this to be the case between 0.05 and 0.4, where 0.05 appears smoother. However, there is little variation regardless, which is likely due to the case of using an exponential distribution, suggesting that it is more stable against different probabilities.

Inter- and Intra-Trajectory Historical Averaging

Figures 9 and 10 show our results comparing inter and intra historical averaging (HA). In particular, we note that not only does intra-jectory HA experiments perform worse, they don't converge to an equilibria and regularly learn. As intra-HA samples more often, it is likely to take longer to reach

Figure 7: Historical averaging with probability of 0.05 under an exponential sampling distribution

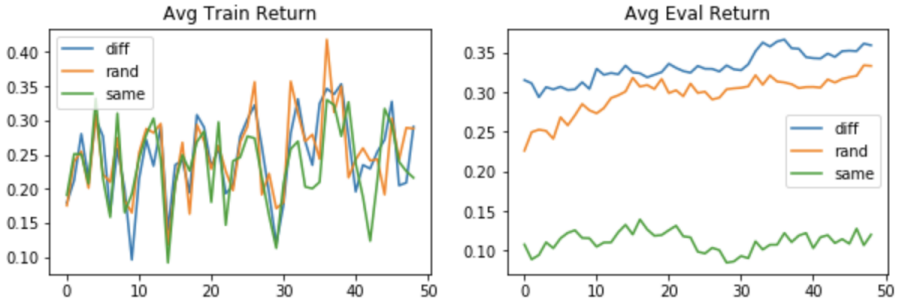
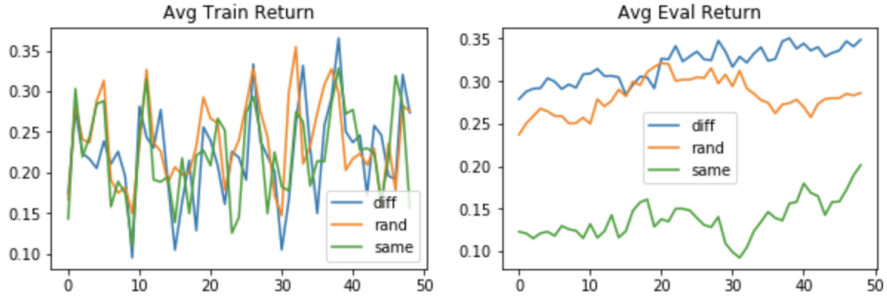


Figure 8: Historical averaging with probability of 0.4 under an exponential sampling distribution



an equilibria and as such would require more teaching iterations. We also see that the agent has a diminished ability to undertake tasks different from what is training on, as evident by the evaluation graph in Figure 10. This is likely due to the idea that since the policies are changing throughout the trajectory, it is difficult for the agent to learn while changing policies and have their rewards that reflect this.

Figure 9: Inter-Historical averaging

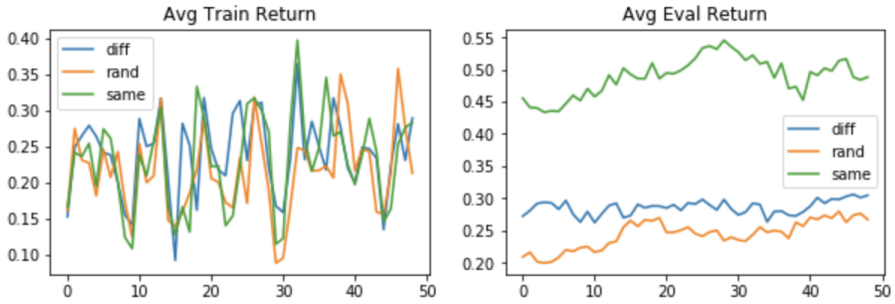
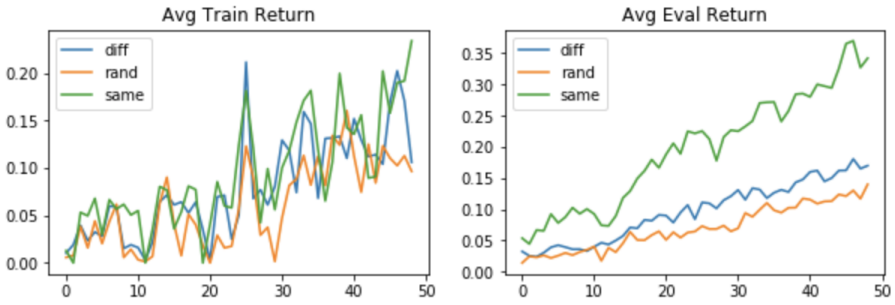


Figure 10: Intra-Historical averaging



5 Discussion

5.1 The Effectiveness of Historical Averaging for Self-Play

Our results show that there is similar performance between agents that have a historical self-play scheme than those with a regular self-play scheme. Self-play itself is a strong tool to help an agent learn an environment better, however we have no guarantees of convergence to a Nash equilibria. Though there are various setups to consider, there are many intuitive and simple methods for choosing how to undergo historical averaging, such as using an exponential sampling distribution to favour more recent weights, and to use inter historical averaging to play more accurately against past policies. With the additional convergence guarantees associated with historical averaging, we see that this often forgotten step in adversarial training is potentially worthwhile and is not detrimental to the learning of the agent.

5.2 Setting Up Historical Averaging

Our experiments have shown how historical averaging performs under a variety of setups, providing an intuition for how a system designer should choose their "historical" self-play scheme. In particular, we note that using inter-trajectory HA and an exponential distribution is likely to induce similar performance to traditional self-play at a potential slight cost of exploration. We also note that for the probability of historical averaging, an exponential distribution is more robust against different probabilities than a uniform probability distribution would be. As such, using a combination of these options can be a good starting place for implementations of historical averaging.

5.3 Future Work

5.3.1 Stochastic Fictitious Play

As we consider players to only play "pure" actions against one another, characterized by the player having a single policy rather than providing an action probabilistically chosen from a set of policies, and thus not using a mixed response, there is no guarantee of convergence to a mixed Nash equilibrium. So in setting where there mixed strategies available, it is suggested in the theory for fictitious play that this can provide convergence guarantees to mixed equilibria by inducing "smoothness" of the play itself. This can be accomplished by adding noise to the reward for each agent, which creates a game of incomplete information. This introduces a new form of fictitious play called "stochastic fictitious play", and a future direction of this work is to analyse how effective this method is.

5.3.2 Continuous Environments

In this work, we consider a simple grid world environment with a door and key object that allows for more difficulty in training the RL agent. However, this is an environment with a discrete state-action space, so further work in applying self-play to continuous settings would be important to see how well this idea can extend to more difficult environments. In particular, this kind of learning would be useful in robotics applications with, for example, learning to unlock a door. In this case, it is possible for a teaching agent can setup tasks to do with the individual motions of unlocking a door, such as picking up a key, placing a key in the keyhole, turning the key, etc., all of which are incrementally difficult tasks, and having an autocurricula approach would be learn this approach of decomposing a problem into smaller sub-problems.

5.3.3 Goal Conditioned Learning

As an extension on self-play, we can use the proposed training goal from the teacher as another input to the student. This method allows a student to reach arbitrary goal states and enables a deeper understanding of the environment. Goal conditioned learning in self-play might allow a student to accomplish more complex tasks. Perhaps, the implementation of goal-conditioned in historical self-play might be useful for learning policies, understanding hierarchical reinforcement learning, and encouraging exploration.

6 Conclusion

Our work introduces an augmented adversarial learning scheme called 'Historical Self Play' (HSP). For comparison, we first implement a version of asymmetric self-play generalized to the expected reward – which was able to effectively pre-train an agent to learn its environment and outperforms an agent trained with a standard learning procedure. We propose the idea of HSP by using theory from fictitious play to introduce a historical averaging step into regular self-play, providing us with convergence guarantees to pure Nash equilibria. We show that there is little difference in the learning of the agent with HSP as opposed to just using self-play, suggesting that this new method to be practical and worthwhile if using adversarial approaches to autocurricula generation. We conclude by suggesting future directions to improve HSP by considering stochastic rewards and to promote the use of self-play, particularly in applying the same method to continuous settings where there are more use cases, and in goal-conditioned learning, which can take advantage of the task-proposal portion of self-play.

References

Justin Fu, John D. Co-Reyes, and Sergey Levine. EX2: exploration with exemplar models for deep reinforcement learning. *CoRR*, abs/1703.01260, 2017. URL http://arxiv.org/abs/1703.01260.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015. ISSN 00280836. URL http://dx.doi.org/10.1038/nature14236.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2016. URL http://dblp.uni-trier.de/db/conf/iclr/iclr2016.html#LillicrapHPHETS15.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 41–48, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553380. URL http://doi.acm.org/10.1145/1553374.1553380.

Joel Z. Leibo, Edward Hughes, Marc Lanctot, and Thore Graepel. Autocurricula and the Emergence of Innovation from Social Interaction: A Manifesto for Multi-Agent Intelligence Research. *arXiv e-prints*, art. arXiv:1903.00742, Mar 2019.

Sainbayar Sukhbaatar, Ilya Kostrikov, Arthur Szlam, and Rob Fergus. Intrinsic motivation and automatic curricula via asymmetric self-play. *CoRR*, abs/1703.05407, 2017. URL http://arxiv.org/abs/1703.05407.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

Lisa Lee, Benjamin Eysenbach, Emilio Parisotto, Eric Xing, Sergey Levine, and Ruslan Salakhutdinov. Efficient Exploration via State Marginal Matching. *arXiv e-prints*, art. arXiv:1906.05274, Jun 2019.

Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic Goal Generation for Reinforcement Learning Agents. *arXiv e-prints*, art. arXiv:1705.06366, May 2017a.

Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. Reverse Curriculum Generation for Reinforcement Learning. *arXiv e-prints*, art. arXiv:1707.05300, Jul 2017b.

A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM J. Res. Dev.*, 3(3):210–229, July 1959. ISSN 0018-8646. doi: 10.1147/rd.33.0210. URL http://dx.doi.org/10.1147/rd.33.0210.

Gerald Tesauro. Temporal difference learning and td-gammon. *Commun. ACM*, 38(3):58–68, March 1995. ISSN 0001-0782. doi: 10.1145/203330.203343. URL http://doi.acm.org/10.1145/203330.203343.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, jan 2016. ISSN 0028-0836. doi: 10.1038/nature16961.

Julia Robinson. An iterative method of solving a game. *Annals of Mathematics*, 54(2):296–301, 1951. ISSN 0003486X. URL http://www.jstor.org/stable/1969530.

Johannes Heinrich, Marc Lanctot, and David Silver. Fictitious self-play in extensive-form games. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 805–813. JMLR.org, 2015. URL http://dl.acm.org/citation.cfm?id=3045118.3045205.

Johannes Heinrich and David Silver. Deep reinforcement learning from self-play in imperfect-information games. *CoRR*, abs/1603.01121, 2016. URL http://arxiv.org/abs/1603.01121.

Ulrich Berger. Brown's original fictitious play. *Journal of Economic Theory*, 135:572–578, 02 2007. doi: 10.1016/j.jet.2005.12.010.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. *CoRR*, abs/1602.01783, 2016. URL http://arxiv.org/abs/1602.01783.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.