Studying the Effects of a Teacher's Risk Tolerance on an Interactive Reinforcement Learning Agent

Spyros Orfanos

December 13, 2020

Abstract

Expected utility theory argues that humans act rationally to maximize their individual satisfaction, known as expected utility. Utility is dependent on an individual's risk preference, meaning that rational decision making does not necessarily yield the highest expected reward. However, a reinforcement learning agent's goal is to act in a way such that the expected reward is maximized. This project studies this misalignment of reward and utility by comparing how teachers of varying risk profiles affect the learning of an interactive reinforcement learning agent. The task considered is a simplified 1-D game of golf where the goal is to minimize the number of shots required to complete the hole. There are several hazards throughout the hole, so the player must choose from a set of clubs, each with its own risk-reward trade-off. The agent learns by interacting with the golf environment in addition to demonstrations from a teacher. The arrival times of demonstrations and the total number of demonstrations are varied. Overall, the risk-seeking teacher is beneficial to the agent, resulting in higher rewards and lower risk when compared to an agent that learns without a teacher. The risk-averse teacher is shown to have detrimental long-term effects on the agent's performance.

1 Introduction

Behavioral economics poses the following question: would you choose option A, which always gives you \$800, or option B, which gives you \$1000 85% of the time and \$0 15% of the time. Here, the exact dynamics of the environment are known, yet different individuals choose different actions. In fact, most people choose option A even though option B has a higher expected return [1]. Expected utility theory argues that individuals take actions to maximize their expected utility, which is different from the expected value as it takes into account their risk preference. Generally, humans are risk-averse which may explain why most people choose option A. Further, it is often difficult for us to act to maximize average reward - we may instead take a safe action if we can't "average out" our reward over multiple trials. If this experiment of choosing between A and B was repeated 1,000 times, then it would be wise to choose option B as it yields a higher return 99.998% of the time. However, like most real-world interactions, we don't have the luxury to 'average out' the consequences of our actions - our decisions are final and the consequences cannot be undone. This makes criticizing our own performance challenging, especially since two experts may produce different decision due to their underlying risk profiles. In complex settings involving risk introduced by randomness (e.g.: financial markets, golf, games involving dice), it is usually difficult to distinguish if a policy is attempting to maximize expected utility or expected reward, let alone which approach is optimal. As such, we are not be comparing the skill level of two human teachers, but instead their risk aversion level. Difficult to evaluate a person's performance without replay - easier to classify as risky or safe.

The goal of a reinforcement learning (RL) agent is to take actions that maximize the expected reward – in the above example, an optimal agent would always choose option B. This contrasts human decision making in two ways. Firstly, the agent's goal is to maximize expected *reward*, not utility. Unless a specified utility function is taken into account when designing the rewards, there is a misalignment between how an optimal agent acts and how a rational individual may act. Secondly, by using a simulated environment, an RL agent learns over the course of many, many interactions - the agent acts to maximize *expected* reward.

In tasks where the optimal behaviour is unknown and cannot be readily tested, it can be easier to classify behaviour according to its risk level. This project studies this misalignment of objectives outlined above by examining how the risk preference of a teacher can influence an interactive RL agent that learns from demonstrations (LfD) [2,3]. We compare three agents: the first which learns on its own (baseline), the second agent is aided with demonstrations from a risk-seeking teacher, and the third is aided with demonstrations from a risk-averse teacher. Several experiments are conducted where the arrival rate of demonstrations and availability of the teacher are varied. The performance (measured by return) and its risk level (measured by a well-defined risk measure) are evaluated. The agent is trained using the proximal policy optimization algorithm, and its task is to minimizing the score in a simplified 1-D hole of golf, outlined in the next sections.

2 Background

2.1 Reinforcement Learning

Reinforcement learning [4] focuses on training an agent to make a sequence of decisions that maximize a long-term reward. The agent learns from experience by way of a Markov decision process (MDP), which defines the agent's interaction cycle. An MDP is defined by the tuple $\{S, A, R, p\}$ which consists of the set of states, the set of actions, the reward function, and the transition probability matrix. At each step, the agent observes a state s and samples an action a according to its policy, a probability distribution denoted $\pi(a|s)$. The agent receives a reward signal, r, and there is a transition to the next state, s', as per the dynamics of the model (i.e. the transition probability matrix). If the task eventually terminates, this interaction gives rise to the sequence $\{S_0, A_0, R_1, S_1, A_1, ..., R_T, S_T\}$, which is called an *episode*.

The value function, $v_\pi(s)$, is defined as the expected return (sum of future discounted rewards) given the agent is currently in state s and acts according to a policy π. That is,

$$v_\pi(s) \;=\; \mathbb{E}_\pi[G_t|S_t = s] \;=\; \mathbb{E}_\pi\left[\sum_{k=0}^{T-t-1} \gamma^k R_{t+k+1}\middle|S_t = s\right]$$

An optimal policy, denoted π^*, is one that maximizes $v_\pi(s)$ for all states, $s \in S$. By experiencing many episodes, an RL agent learns which actions produce the highest return, and adjusts its policy accordingly. A model-free RL agent does not attempt to learn the transition probabilities; instead it relies on its experience to estimate the value function. The value function can be estimated with the Monte Carlo (MC) target, which is the sample average returns observed after being in state s. The MC target is unbiased; however, it suffers from high variance. For the temporal difference (TD) approach, $v_\pi(s)$ is estimated by $R_{t+1} + \gamma v_\pi(s)$ - this target is biased but has lower variance than the MC target. The n-step return and the λ-return both provide a way to move between these two targets. The λ-return is considered in the next section.

2.2 Proximal Policy Optimization

Proximal policy optimization (PPO) [5] is an actor-critic method, meaning it maintains two networks: an actor network, and a critic network. The actor network, denoted $\pi_\theta(a|s)$, represents the policy π and is used to interact with the environment. The critic network, denoted $\hat{v}_\psi(s)$, approximates the value function $v(s)$ and is used to evaluate the actor network, which learns from this criticism. PPO is an on-policy method, meaning the policy that is used to interact with its environment is varied. PPO is also an online leaning algorithm, meaning that once an experience is used to make a learning update, it is discarded. This implementation uses λ-returns, G_t^λ, to compute the advantage estimates, $\hat{A}_t$, and as a target for $\hat{v}_\psi(s)$. The surrogate loss objective clips the estimated advantage to prevent the policy from changing drastically (hence proximal policy), thereby improving the stability of the actor. The surrogate loss objective and implementation of PPO are presented in the following pseudo-code [6].

Algorithm 1: PPO Pseudo-code

```
Initialize network weights $\theta$, $\psi$
for each episode $k = 1, ...$ do
    Experience: $\{S_t^k, A_t^k, R_{t+1}^k\}_{t=0}^{T_k-1}$ by acting according to $\pi_\theta$
    Compute: $\{\hat{v}_\psi^k(s_t)\}_{t=0}^{T_k-1}$, $\{G_t^{\lambda,k}\}_{t=0}^{T_k-1}$
    if $k$ mod K = 0 then
        $\theta' \leftarrow \theta$, $\psi' \leftarrow \psi$
        Advantage Estimate: $\hat{A}_t = G_t^\lambda - \hat{v}_\psi(s_t)$ for each t,k in the entire batch.
        Normalize $\hat{A}_t$ over the batch
        for each E epochs do
            Shuffle the batch, slice into N mini-batches
            for each mini-batch do
                $l(\theta', \psi')$ = -mean$\Big[\zeta_t(\theta, \theta', \psi)\Big]$ + mean$\Big[\Big(G_t^\lambda - \hat{v}_{\psi'}(s_t)\Big)^2\Big]$
                where $\zeta_t(\theta, \theta', \psi) = \min\Big[\frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)}\hat{A}_t, clip\Big(\frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)}, 1-\epsilon, 1+\epsilon\Big)\hat{A}_t\Big]$
                Update network weights $\theta', \psi'$ using Adam with loss $l(\theta', \psi')$
        $\theta \leftarrow \theta'$, $\psi \leftarrow \psi'$
```

2.3 Incorporating a Teacher

In this paper, the agent learns by interacting with the environment and from demonstrations given by a teacher. Every F episodes, the agents observes a teacher's episode, and records the experience to its buffer. A demonstration is equivalent to the teacher forcing the agent to follow its policy for an episode. For a given run, a teacher demonstrates an episode exactly B times. This means for a fixed B, if demonstrations are presented more frequently, the teacher becomes unavailable at an earlier time of the agent's learning process. This interaction is implemented as per the pseudo-code below.

Algorithm 2: Human-in-the-Loop PPO

```
Initialize network weights $\theta$, $\psi$
for each episode $k = 1, ...$ do
    if $k$ mod F = 0 and B > 0 then
        Experience: $\{S_t^k, A_t^k, R_{t+1}^k\}_{t=0}^{T_k-1}$ by observing teacher's demonstration
        Compute: $\{\hat{v}_\psi^k(s_t)\}_{t=0}^{T_k-1}$, $\{G_t^{\lambda,k}\}_{t=0}^{T_k-1}$
        B = B - 1
    else
        Experience: $\{S_t^k, A_t^k, R_{t+1}^k\}_{t=0}^{T_k-1}$ by acting according to $\pi_\theta$
        Compute: $\{\hat{v}_\psi^k(s_t)\}_{t=0}^{T_k-1}$, $\{G_t^{\lambda,k}\}_{t=0}^{T_k-1}$
    if $k$ mod K = 0 then
        Perform PPO update
```

2.4 Risk Measures

To quantify a policy's level of risk, the tail-value-at-risk (TVaR) [7] of the rewards it produces will be studied. TVaR is a risk measure that is commonly used in actuarial science to study losses at the tail-end of a distribution, and is interpreted as the expected loss given the loss is 'significantly large'. A 'large' loss at the 1-α significance level is defined as the quantile Q$_\alpha$, where $Pr(X < Q_\alpha) = \alpha$. The TVaR at the 1-$\alpha$ significance level for a random variable X is defined as TVaR$_\alpha(X)$ = E[$X|X \leq$ Q$_\alpha$]. In this paper, 'X's TVaR' is to be understood as the TVaR of the random variable G$_0$ (the undiscounted episodic return) produced by X's policy. The TVaR at the 99%, 95%, and 90% significance level is studied for the final 5,000 episodes, long after the teacher has stopped advising the agent. This will reveal if a teacher can have a long-term influence on the agent's risk level.

3 Experiment Design

3.1 Golf Environment

Two players of equal capabilities may take different approaches to the game of golf. A risk-averse golfer would rather ‘play it safe’ and opt to hit multiple shots with a club that has a predictable outcome. A risk-seeking golfer may opt to take a risky shot that may carry the ball close to the flag but may land the ball in the water. Golfers can approximate the range of shots that each club can produce, and hence can be considered model-based learners. This environment allows us to address the misalignment of reward and utility in a more complex case, where the optimal strategy unclear to a human.

The game of golf can be formulated as an MDP: the agent (golfer) must decide which action (club) to choose given the current state (position of the ball). After each action, a reward (stroke) is incurred and there is a transition to the next state (new position of ball). This process repeats until a terminal state is reached (the ball is holed). Further, the transition probabilities can be calculated if the distribution of each club’s distance is known. If the ball lands in a water hazard or goes past the limits of the hole, the shot must be replayed and the player incurs an additional one stroke penalty (i.e. reward of -2), as per the rules of golf. Otherwise, a reward of -1 is given. As such, maximizing the reward is equivalent to minimizing score, which is precisely the objective of golf. More specifically, we have the following:

- $S = \big[0, 140\big] \cup \big[190, 270\big] \cup \big[300, 405\big]$ (i.e. the ball is in play, on the fairway)
- $S_0 = [0,\ 10]$ (i.e. the starting state is randomly chosen within the first ten yards)
- $S_T = [390,\ 420]$ (i.e. the episode terminates once the ball is within 15 yards of the hole)
- $A = \{0, 1, 2\}$ (i.e. the set of clubs to choose from)
- Once action A_t is selected, the resulting shot distance is obtained by drawing a sample of the conditional random variable $\mathrm{D}|A_t$ from the following Gaussian distributions:
  - $\mathrm{D}|(A_t = 0) \sim \mathcal{N}(180, 20)$
  - $\mathrm{D}|(A_t = 1) \sim \mathcal{N}(105, 10)$
  - $\mathrm{D}|(A_t = 2) \sim \mathcal{N}(55, 7)$
- $R_{t+1} = -1 - \mathbb{1}_{PenaltyIncurred}$

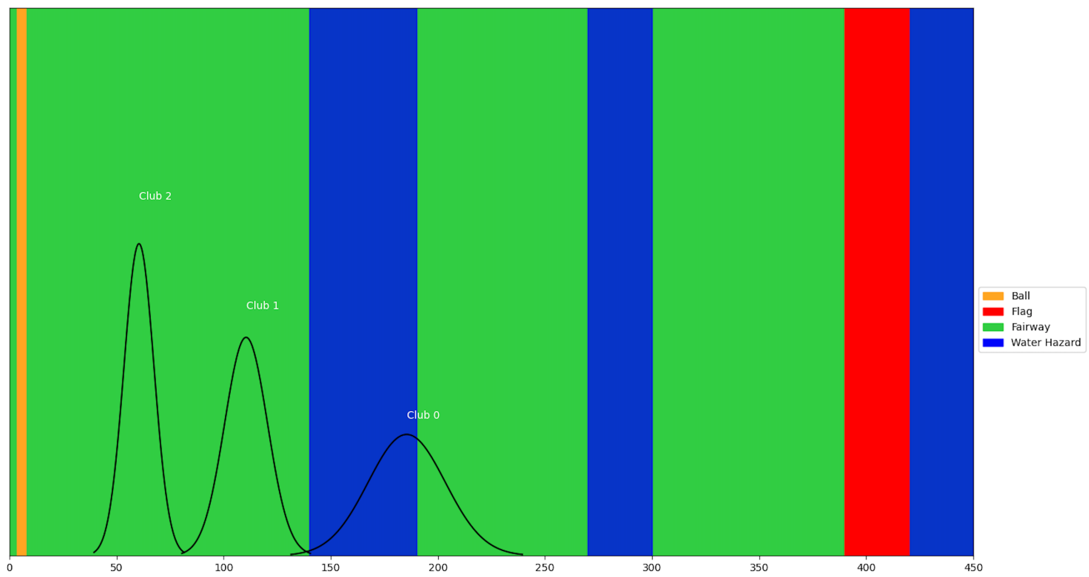

Figure 1: Golf Hole superimposed with Distribution of Each Club’s Distance

3.2 Golf Teachers

The risk-seeking (risky) teacher chooses a club that can carry the ball closest to the flag, even if there is risk of going in the water. The risk-averse (safe) teacher hits a shot that has a very low chance of landing in the water, even if it means sacrificing distance and proximity to the flag. The exact policies of each teacher can be found in Appendix A.

Although it is not always possible to evaluate a human's performance, these two policies were tested for 500,000 episodes each in the simulated golf environment. The risky teacher averaged a return of -5.95 ± 4.30 and the safe teacher averaged a return of -4.42 ± 0.977. The risky teacher is able to achieve scores of -2 or -3 37% of the time; however, 61% of the time this policy resulted in a return of -6 or worse. The safe teacher is unable to produce a score of -2 and only 11% of the time can score -3, however; 95% of the time achieves a score of -5 or better. More details are provided in appendix A. The sample TVaR of the two teachers' returns are summarized in the table below. As expected, the risky teacher's TVaR is lower than the safe teacher's, meaning the worst return it produces are worse than the worst return produced by the safe teacher.

α	Risky	Safe
1%	-26.7	-11.1
5%	-20.4	-8.17
10%	-16.2	-6.93

Table 1: Sample TVaR of Teachers' Returns

4 Results and Discussion

Recall that a teacher gives a demonstration every F episodes for a total of B episodes. Six configurations of F × B are studied by choosing F from $\{1, 5, 15\}$ and B from $\{600, 1500\}$. Each configuration was run for 30,000 episodes and averaged over 10 independent runs. Two separate multi-layered perceptrons of two hidden layers of size 32 are used to represent the actor and critic networks. The hyperbolic tangent function is used as the activation functions in both networks. The Adam optimizer with a learning rate of 0.001 was used. A learning update consisted of 5 epochs (E=5), 10 mini-batches (N=10), and a batch size of 30 episodes (K=30). We used trust region parameter ϵ=0.2, discount factor γ=1, and credit assignment λ = 0.95. The plots show the number of training batches versus the average return in said training batch. The dotted orange and green lines represent the average return of the risky and safe policies, respectively. The statistics regarding the agent's performance measures (i.e. mean, standard deviation, and TVaR) are calculated using the final 5,000 episodic returns from all 10 runs. Without any aid from a teacher, the agent averaged a return of -4.37 ± 2.02 over the last 5,000 episodes. This agent will serve as the baseline for this experiment.

4.1 F=1, B=600

Here, the agent observes the teachers' experience for the first 600 episodes, and learns on its own thereafter. With demonstrations from the risky teacher, the agent improved slightly and averaged a score of -4.28 ± 1.92. Learning from demonstrations given by the safe teacher, the agent averaged a return of -4.56 ± 2.49.

α	None	Risky	Safe
1%	-14.8	-13.6	-18.5
5%	-9.85	-9.91	-11.8
10%	-8.71	-7.53	-9.25

Table 2: Agent's Sample TVaR under Various Teachers' Instruction

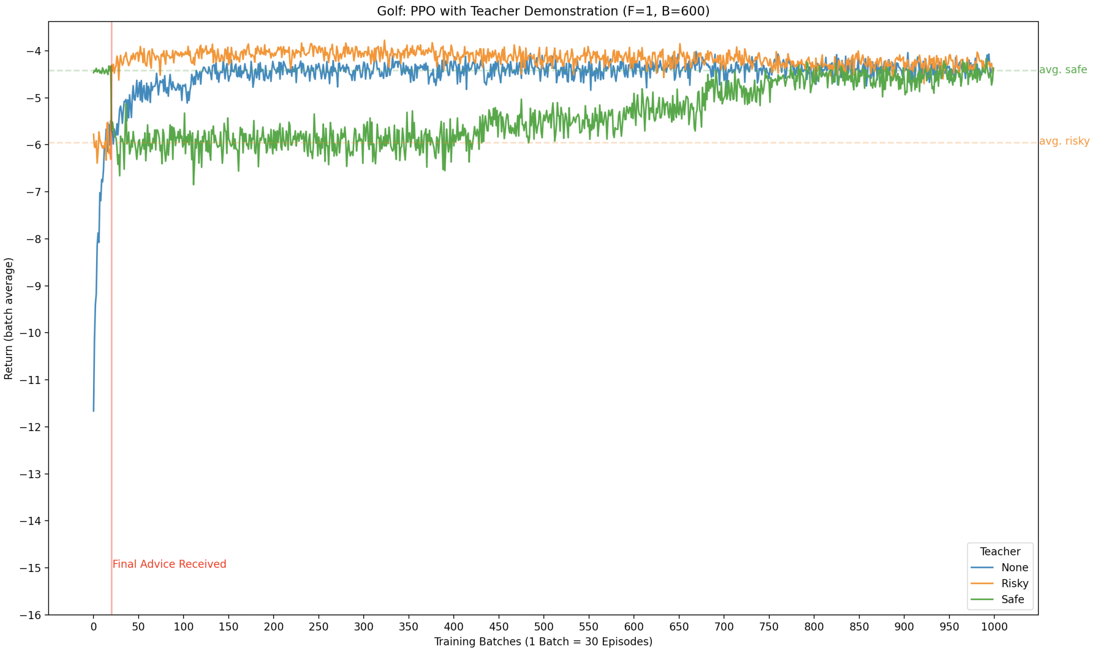

Figure 2: Comparing Average Batch Returns - averaged over 10 Runs (F=5, B=600)

Immediately following the teachers' demonstrations, there were two surprising jumps in the agent's average return. Once the agent that learned from risky demonstrations was able to act on its own (after 20 batches), it immediately began to average a high return of -4.28. It maintained an advantage over the the baseline for the first 750 batches, after which all three agents averaged similar scores. This agent produced roughly the same TVaR statistics as the baseline. Surprisingly, the safe teacher's demonstrations resulted in an average return lower than the baseline. It remained at this low performance level for approximately 10,000 episodes, and then it slowly improved to a policy that was roughly on-par[1]with the other two agents. The safe teacher did not influence the agent to take more 'safe' trajectories, as noted by the increase in TVaR for all significance levels.

Until the risky teacher lands past the second water hazard, it selects action 0 which produces the highest variation of outcomes. Hence, it exposes the agent to a wide variety of states, effectively forcing exploration. When the adverse consequences of exploratory actions are catastrophic (e.g. falling of the cliff results in reward of -100 and episode termination in Cliff Walking), this teacher may not be a suitable instructor. However, in golf, the consequence of landing in a water hazard is much less severe: it only result in an additional one stroke penalty and the ball to be replaced to where it was hit from. Further, since the risky teacher sometimes produces very poor results, it can also help the agent learn what **not** to do. Conversely, the safe teacher exposes the agent to a much more limited range of states due to the clubs a safe teacher opts to hit. This discourages exploration, so there is less marginal value for observing multiple demonstrations of the safe teacher. Perhaps this explains why the safe teacher was not very helpful.

4.2 F=1, B=1500

Here, a similar but more exaggerated trend can be observed: the risky teacher improves the agent's performance while the safe teacher worsens it. Under the risky teacher the agent averaged a score of -4.06 ± 1.92, and under the safe teacher averaged a return of -5.79 ± 4.11. When learning from the safe teacher's demonstrations, the agent seems to be trapped in a local minimum of approximately -5.79, and produces an average return significantly worse than the baseline. The risky teacher slightly improved the agent's TVaR for all significance levels, while the safe teacher had the opposite effect.

[1] Golf pun

α	None	Risky	Safe
1%	-14.8	-15.6	-25.0
5%	-9.85	-9.24	-18.3
10%	-8.71	-7.65	-14.7

Table 3: Sample TVaR of Agent's Returns under Various Teachers' Instruction

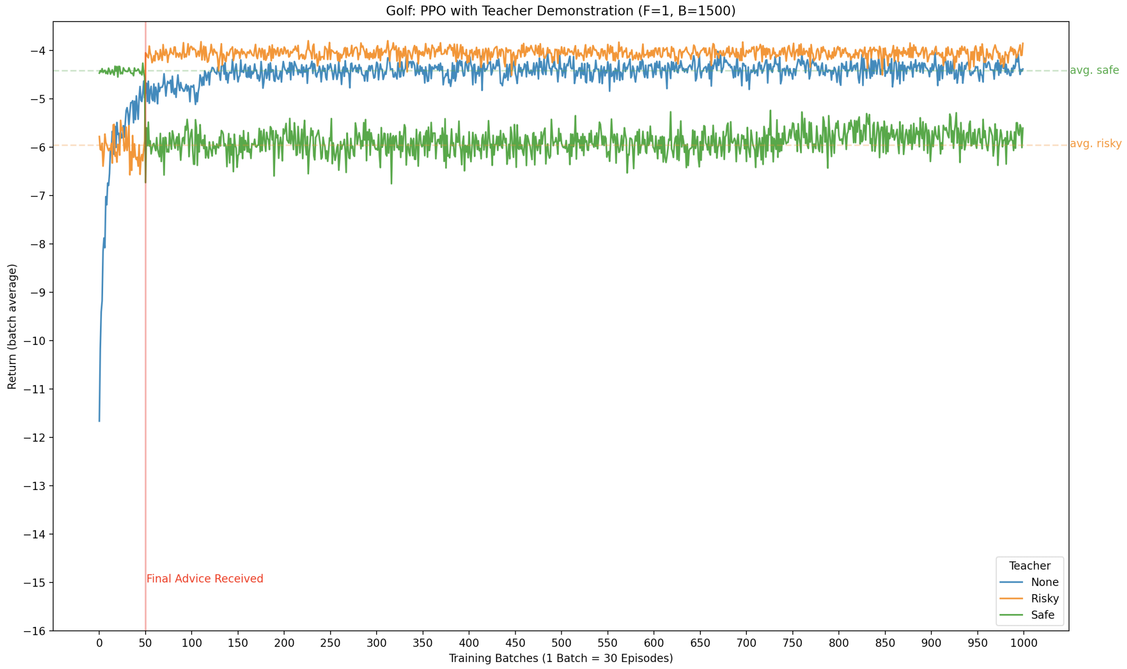

Figure 3: Comparing Average Batch Returns - averaged over 10 Runs (F=5, B=600)

4.3 F=5, B=600

In this experiment, the agent observes a teacher's demonstration every 5 episodes for a total of 600 episodes. The teacher's final demonstration was presented on the 3,000th episode, which is in the 100th learning update. As was the case previously, the risky teacher is helpful to the agent and the safe deacher is detrimental. With demonstrations from the risky teacher, the agent improved and averaged a score of -4.04 ± 1.84. Finally, with help from the safe teacher, the agent achieved an average return of -5.30 ± 3.61. As observed in Figure 4, both teachers were helpful to the agent in the first 50 batches. This can be expected as both teachers have prior knowledge of where the hazards are, how the club's distances are distributed, and what the rules of golf are.

α	None	Risky	Safe
1%	-14.8	-14.0	-23.0
5%	-9.85	-9.21	-16.8
10%	-8.71	-7.66	-12.5

Table 4: Sample TVaR of Agent's Returns under Various Teachers' Instruction

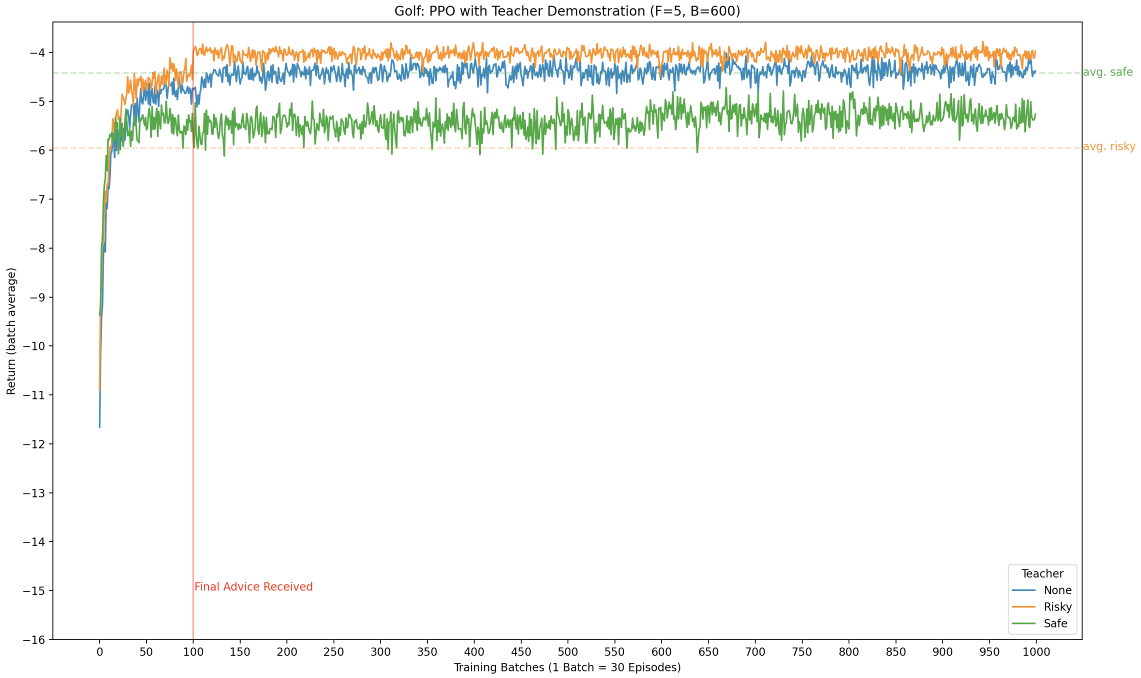

Figure 4: Comparing Average Batch Returns - averaged over 10 Runs (F=5, B=600)

4.4 F=5, B=1500

There was not a significant change when increasing the number of demonstrations to 1500 episodes. The results can be found in Appendix B.

4.5 F=15, B=600

Here, the agent learns from a demonstration every 15 episodes for a total of 600 episodes. The teacher's final demonstration is presented on the 3,000th episode, which is in the 100th batch. Without any teacher, the agent averaged a return of -4.89 ± 2.74. With demonstrations from the risky teacher, the agent improved and averaged a score of -4.04 ± 1.72. Finally, with help from the safe teacher, the agent achieved an average return of -4.84 ± 2.96. Here, effect of a teacher is still considerable, but less impactful on the agent's performance than the previous configuration was.

α	None	Risky	Safe
1%	-14.8	-14.6	-21.1
5%	-9.85	-9.03	-13.6
10%	-8.71	-7.58	-11.2

Table 5: Sample TVaR of Agent's Returns under Various Teachers' Instruction

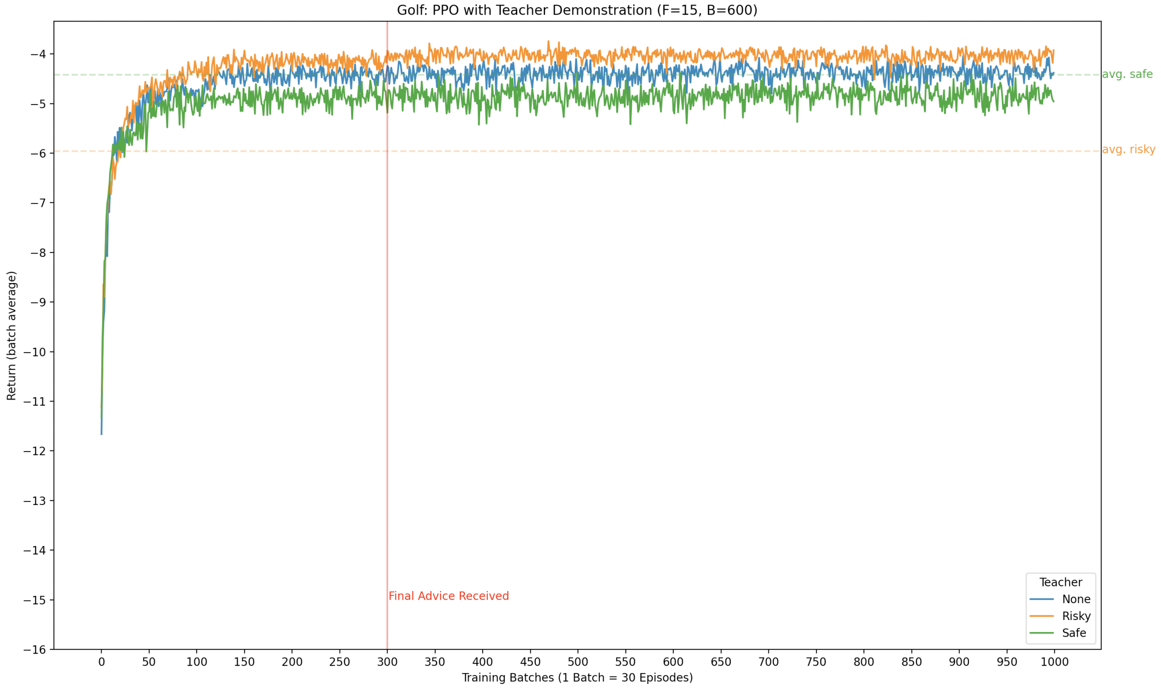

Figure 5: Learning Curves averaged over 10 Runs (F=5, B=1500)

4.6 F=15, B=1500

There was not a significant change when increasing the number of demonstrations to 1500 episodes. The results can be found in Appendix C.

5 Related Work

Geibel and Wysotzki [8] take a more direct approach to incorporating risk in a constrained MDP framework. The agent's goal is to maximize reward subject to a risk constraint. Liu et al. [9] consider an RL framework where the task is to maximize an expected utility function that can penalizes high variance of rewards. These papers do not consider if input from a teacher can influence an RL agent's risk profile, which is studied in this project.

6 Conclusion

This work explores the impact of a teacher's risk tolerance on an interactive RL agent that learns from demonstration. The arrival rate (i.e. how often a teacher presents a demonstration) and the total number of demonstrations were varied, with the former having a more profound impact on the agent's learning. Overall, when compared to the baseline agent, the risk-seeking teacher improved the average episodic reward and TVaR risk measure of the baseline agent, while the risk-averse teacher was detrimental to the agent's learning. The risk-averse teacher was unable to influence the agent to act 'safely', as observed by large negative TVaR's. This approach was not suitable to influence the agent's risk profile; instead, providing the agent with more direct feedback, such as in the TAMER+RL framework [10], may offer more significant results.

A possible concern was that the trust region parameter, ϵ, was too small and prevented the agent from learning the safe teacher's policies. To test this, a higher trust region parameter of ϵ=0.5 was also considered; however, a similar pattern was observed. Future research could study this project's framework using a higher learning rate and an actor-critic algorithm where the change to the policy is not as restricted. In addition, the policies of the two teachers, as well as near-optimal policy found by an agent who learned from risky demonstrations are presented in Appendix A. Note that this near optimal policy is simple, yet quite different from the two teachers' policies.

Further research can consider the action advising frameworks proposed by Taylor et al. [11, 12]. For example, requesting advice when there is a high degree of uncertainty (e.g. the actor network exhibits a high entropy). Further work may consider choosing from multiple teachers, each with some cost of advice.

7 Acknowledgements

I would like to thank Dr. Matt Taylor for providing feedback, advice, and support throughout this project. This project was implemented in Python3 with PyTorch.

References

[1] S. Okasha. Rational Choice, Risk Aversion, and Evolution. *The Journal of Philosophy*, 104(5):217-235, 2007.

[2] B. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

[3] M. Taylor, H. Suay, and S. Chernova. Integrating reinforcement learning with human demonstrations of varying ability. *Autonomous Agents and Multiagent Systems (AAMAS)*, 2011.

[4] R. Sutton and A. Barto. *Introduction to Reinforcement Learning.* MIT Press, 1998.

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint* arXiv:1707.06347, 2017.

[6] R. Mahmood. CMPUT 653. Class Lecture, Topic: "Continuous Control: Reinforce to Batch Actor-Critic to PPO". Department of Computing Science, University of Alberta, Edmonton, AB. Sept. 30, 2020.

[7] S. Klugman, H. Panjer, and G. Willmot. *Loss Models: From Data to Decisions.* WILEY, 2012.

[8] P. Geibel and F. Wysotzki. Risk-Sensitive Reinforcement Learning Applied to Control under Constraints. *Journal of Artificial Intelligence Research*, 24(5):81-108, 2005.

[9] Y. Liu, R. Goodwin, and S. Koenig. Risk-averse auction agents. In *Proceedings of the Second International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, (03):353–360, 2003.

[10] W.B. Knox and P. Stone. Reinforcement Learning from Simultaneous Human and MDP Reward. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2012.*

[11] L. Torrey and M. Taylor. Teaching on a Budget: Agents advising agents in reinforcement learning. In *Proceedings of 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 1053-1060, 2013.

[12] F.L. Da Silva, P. Hernandez-Leal, B. Kartal, and M. Taylor. Uncertainty-Aware Action Advising for Deep Reinforcement Learning Agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(4):5792-5799, 2020.

Appendix A Some Notable Policies

Algorithm 3: Risky Policy

```
Let $s_t$ be the position of the ball at time $t$. The risky teacher acts according to the following deterministic policy.
if $s_t \leq 260$ then
    $A_t = 0$
else if $s_t \leq 325$ then
    $A_t = 1$
else
    $A_t = 2$
```

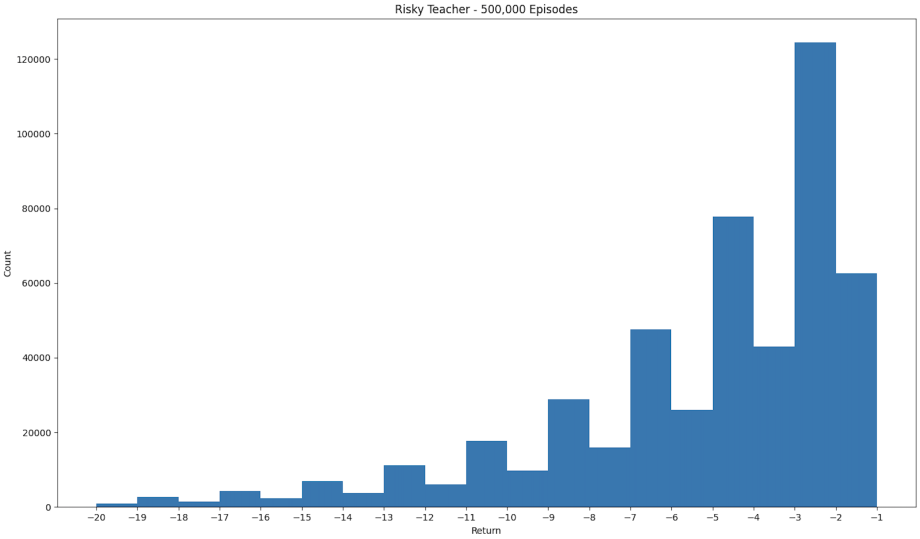

Figure 6: Histogram of Risky Teacher's Returns (censored below at -20, N=500,000). The cyclic distribution is likely a result of incurring an additional one stroke penalty

Algorithm 4: Safe Policy

```
Let $s_t$ be the position of the ball at time $t$. The safe teacher acts according to the following deterministic policy.
if $s_t \leq 20$ then
    $A_t = 1$
else if $s_t \leq 70$ then
    $A_t = 2$
else if $s_t \leq 93$ then
    $A_t = 0$
else if $s_t \leq 155$ then
    $A_t = 1$
else if $s_t \leq 212$ then
    $A_t = 0$
else if $s_t \leq 300$ then
    $A_t = 1$
else
    $A_t = 2$
```

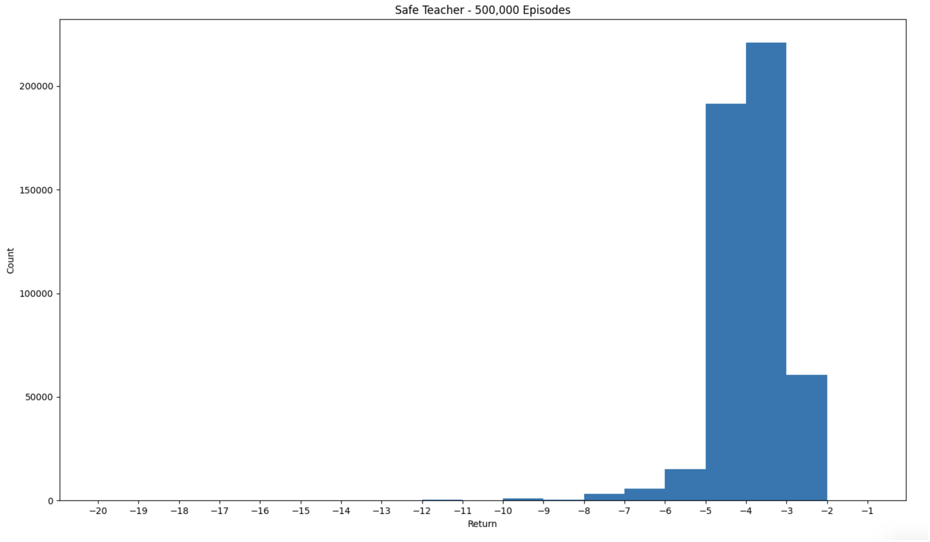

Figure 7: Histogram of Safe Teacher's Returns (N=500,000)

The following is the final policy was taken from Run #8 where F=1 and B=600. This policy was then tested for 500,000 episodes and averaged a return of -3.998 ± 1.71. The sample TVaR's at the 1%, 5%, and 10% significance levels are -13.8, -7.58, and -6.40, respectively. This policy is suspected to be very close to the optimal policy.

Algorithm 5: An Optimal Policy?

```
Let $s_t$ be the position of the ball at time $t$. The risky teacher acts according to the following
deterministic policy.
**if** $s_t \leq 130$ **then**
    $A_t = 1$
**else if** $s_t \leq 255$ **then**
    $A_t = 0$
**else**
    $A_t = 2$
```

Appendix B Results for F=5, B=1500

With demonstrations from the risky teacher, the agent averaged a score of -4.04 ± 1.80. Finally, with help from the safe teacher, the agent achieved an average return of -5.62 ± 4.01.

α	None	Risky	Safe
1%	-14.8	-14.9	-25.1
5%	-9.85	-9.16	-16.9
10%	-8.71	-7.63	-14.8

Table 6: Sample TVaR of Agent's Returns under Various Teachers' Instruction

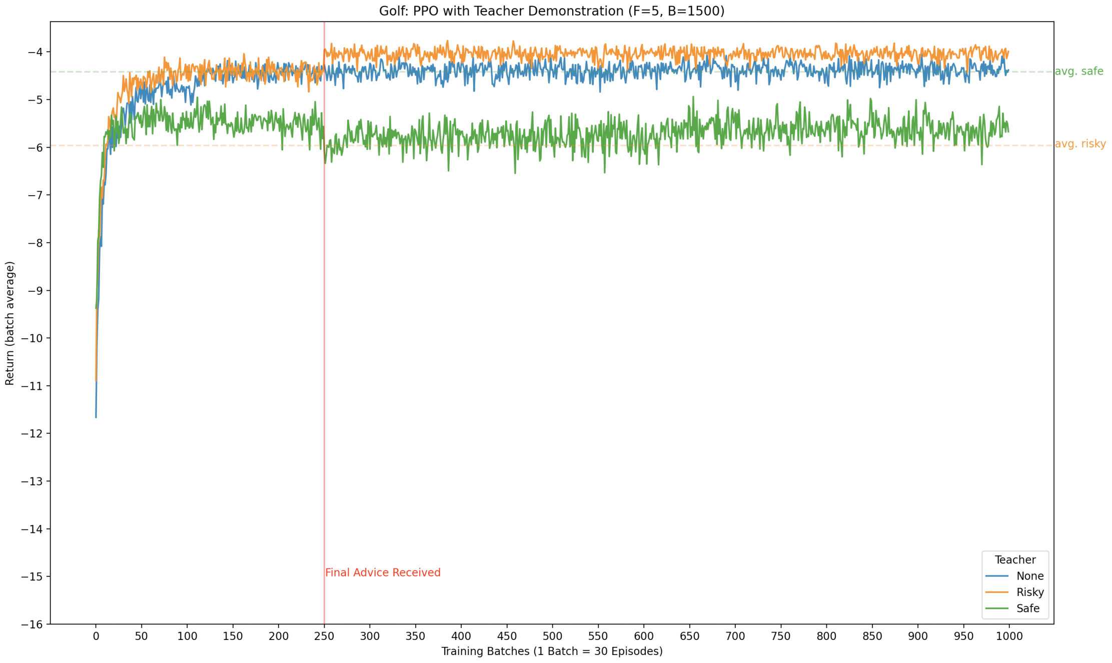

Figure 8: Comparing Average Batch Returns - averaged over 10 Runs (F=5, B=1500)

Appendix C Results for F=15, B=1500

With demonstrations from the risky teacher, the agent averaged a score of -4.03 ± 1.72. Finally, with help from the safe teacher, the agent achieved an average return of -4.83 ± 2.93.

α	None	Risky	Safe
1%	-14.8	-14.6	-21.0
5%	-9.85	-9.17	-13.5
10%	-8.71	-7.63	-11.1

Table 7: Sample TVaR of Agent's Returns under Various Teachers' Instruction

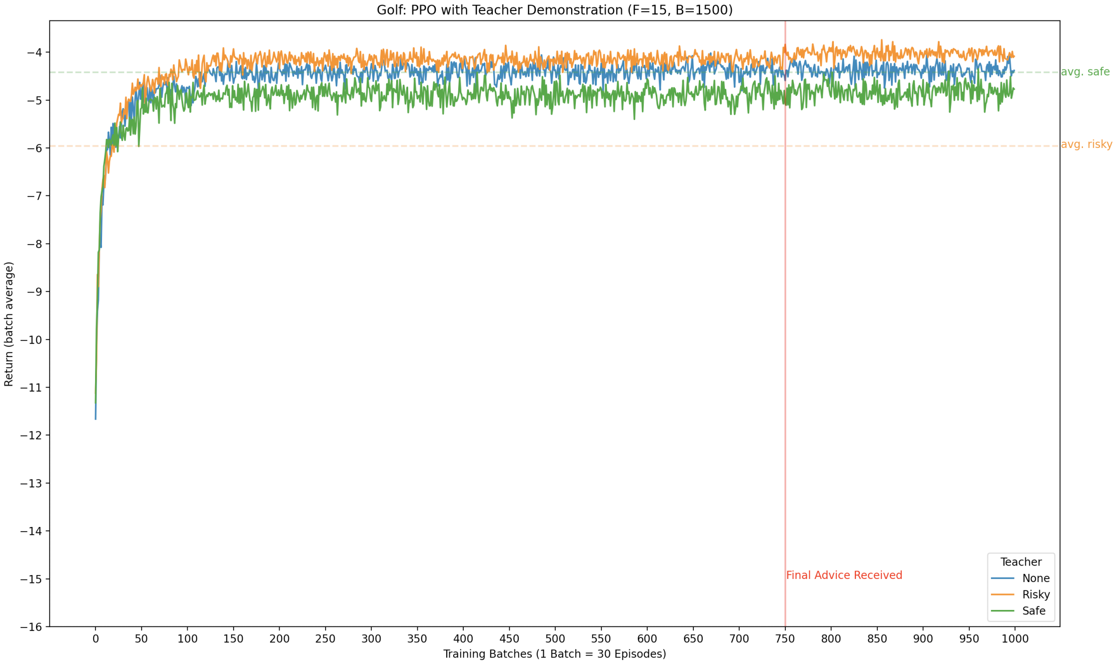

Figure 9: Comparing Average Batch Returns - averaged over 10 Runs (F=5, B=1500)