Advances in Cognitive Systems 10 (2022) 1-12 Submitted 9/2/2022; published 11/2022

Dynamic Discounting in Decision-Making

Derek Martin DMARTIN7@NCSU.EDU
Arnav Jhala AHJHALA@NCSU.EDU
College of Engineering, North Carolina State University, Raleigh, NC 27606 USA

Abstract

In this paper, we explore self-control in cognitive systems using reinforcement learning. We characterize self-control in decision making based on theories of delay discounting proposed in cognitive psychology. Our model explores self-control through different discount functions and learning rates in a Capture-the-flag (CTF) game where immediate rewards are from flag control and defense actions but overall wins are based on successful return of the flag to the base. We provide empirical results using game playing agents for both static and dynamic discounting cases. Results show that hyperbolic and exponential discounting is effective in decisions that have longer time horizons for rewards, such as flag returns to base in the CTF game. We further show the impact of different combinations of dynamic and static settings for discount and learning rates on self-control in our reinforcement learning agents.

1. Introduction

Human decision making has been studied for decades to understand how background knowledge, age/experience, and self-control factor into decision-making. A notable study being the Marshmallow test (Mischel, 1974). This test provided a method to characterize how delay of gratification (i.e. one's ability to resist the urge to take immediate rewards) in children affected their rewards. The children were given the task of being presented with a marshmallow (or pretzels) by a proctor and asked to wait until their return upon which they would be offered another one. At any time the participants could signal for the proctor to return without the second item. The objective of this study was to see how different tasks (e.g. imagining things similar to the reward, distractions/playing, etc.) affected the children's ability to wait for a better reward. This helped (Mischel, 1974) identify the combination of factors that encourage more self-control. Other such studies that fall into either choice theory or decision theory, have been conducted to study decision-making under uncertainty (Tversky & Kahneman, 1992; Kahneman & Tversky, 2013), delay discounting (Green & Myerson, 2004; Odum, 2011), overconfidence based on task difficulty (Klayman et al. (1999)), and other behaviors that humans exhibit when making decisions or performing tasks. Studies like these help researchers to better model and understand different aspects of human decision-making.

Modern cognitive systems have been developed with reinforcement learning (RL) techniques. They have been proven to be effective for training autonomous agents capable of performing at a human-level (Mnih et al., 2015) in domains like Atari games. Despite these recent advances in reinforcement learning, there is still a major issue that pertain to cognitive systems that remain under studied. Often cognitive systems based upon RL focus on model optimization without explicit considerations of task difficulty and duration with respect to rewards, also referred to as time horizons. This is an issue because RL systems rely on direct interactions with the environment to learn, and if the task difficulty is high and/or the feedback is sparse or biased they are more prone to being overconfident or thinking they are doing better than they actually are (Klayman et al., 1999; Jessup et al., 2008). This overconfidence can lead to impulsive behaviors and prevent them from performing well on tasks with delayed rewards/long horizons.

To begin addressing this topic, we explore how learning and preference for long-time horizon tasks (i.e. a task with a long delay between the initial action and the actual reward) and short-time horizon tasks (i.e. a task with a short or no delay between the initial action and the actual reward) changes a cognitive system's overall behavior for game playing. We design models with different learning rates and discount functions that mirror ones found in human studies in cognitive psychology. We demonstrate the variety and effects of different behaviors learned using a capture-the-flag (CTF) task which is a domain that includes both long and short time horizon tasks within the same game.

2. Discounting in RL

Reinforcement learning problems are typically formalized as Markov Decision Processes (MDPs) denoted with the vector: $< S, A, P, R, \gamma >$. S represents the state space, A the action space, $P : S \times A \times S \rightarrow [0, 1]$ is the state-transition probability, $R : S \times A \rightarrow R$ is the reward function, and $\gamma \rightarrow [0, 1]$ is the discount factor. RL agents aim to optimize a policy, π_θ, parameterized by a parameter vector, θ, that aim to maximize the discounted sum of the expected return:

$$G_t = \sum_{i=0}^{N} \gamma^i r_{i+t} \tag{1}$$

this discounted reward is used to account for its future success.

2.1 Learning in RL

Learning in RL agents is performed via policy gradients or action-value updates (Sutton & Barto, 2018). After an agent reaches a terminal state (e.g. death, win, etc.) or some specified time limit, the agent reviews its performance and attempts to update its policy, π_θ to continuously improve its performance until it finds an optimal policy (i.e. a policy that returns the largest reward in each state). In our experiments we use a policy gradient algorithm called Proximal Policy Optimization (PPO) (Schulman et al., 2017), and we modify the G_t in the policy gradient, $\nabla_\theta J$:

$$\nabla_\theta J = E_{\pi_\theta}\left[\sum_{t=0}^{\infty} \nabla_\theta log \pi_\theta(a_t|s_t) G_t\right] \tag{2}$$

to attempt to capture human-like evolution of the discount function. We incorporate both delay discounting and dynamic discounting to achieve this. The gradient updates the policy by pushing the network weights towards the actions with the highest probability based on discounted sum of

Parameters	Settings
gamma	0.99, 0.95, -
learning_rate	2.5e-4, 5e-3, -
discount function (df)	Geometric, Exponential, Hyperbolic
discount scheme (ds)	Static, Increasing, Confidence
learning rate (lr)	Linear, Decreasing, Confidence
total_timesteps	10,000,000

Table 1. Discounting and learning settings available to modify for our agents.

rewards, G_t. To manage the size of the updates to π_θ, we experiment with three variant of annealing for our learning rate:

$$\alpha_{n+1} = \alpha_0(1 - \frac{n-1}{N}) \tag{3}$$

where α_0 is the initial learning rate, n is the current update iteration, and N is the total number of updates. This is the linear learning rate annealing that we use. We use the decreasing learning rate from François-Lavet et al. (2015), which is represented by:

$$\alpha_{n+1} = 0.98\alpha_n. \tag{4}$$

Finally, we use a novel annealing technique based on the agent's policy and the strength of their preference for their selected actions (π_θ) represented by the following equation:

$$\alpha_{n+1} = max(\pi_\theta(a|s))\alpha_0 \tag{5}$$

where $max(\pi_\theta(a|s))$ returns the probability of the preferred action and we use this to determine big of an update the agent should make to their policy. We discuss this approach further below when we discuss dynamic discounting.

2.2 Delay Discounting

To begin exploring human-like discounting, we first manipulate G_t by replacing the geometric function, V = $\gamma^i r_{i+t}$ with either an exponential function or hyperbolic function. Green & Myerson and Odum both review studies that find these two functions better capture delay discounting found in humans and animals. The exponential function is most popular amongst economists and assumes that the value will decrease by a proportionate amount depending on the delay. Exponential discounting is as follows:

$$V = r_{i+t}e^{-\gamma i} \tag{6}$$

where r_{i+t} is the reward received, γ is the discount rate, and i is the time the reward is received relative to the current state. The hyperbolic, or hyperbolic-like, function attempts to account for variance in choices (i.e. choices that have different rates of reward) (Green & Myerson, 2004). Hyperbolic discounting is:

$$V = \frac{r_{i+1}}{(1+\gamma i)^s} \tag{7}$$

with the only new variable being s, a nonlinear scaling factor (i.e. when s=1 the function is hyperbolic and when s<1 it is hyperbolic-like) which determines how steep the discount curve is (Green & Myerson, 2004; Odum, 2011).

Fedus et al. implemented an agent using a hyperbolic discounting function for their experiments in the Atari Learning Environment (ALE) and Pathworld environments. In Pathworld their agent is trained to be robust to hazards in the environment. Their experiments found hyperbolic discounting performed better than geometric discounting. However, in ALE solely using a hyperbolic discount function did not have a significant impact on the performance of their agent. The difference between their work and ours is we are using a domain with a long-time horizon (i.e. we have to consider delayed rewards), real-time strategy (RTS) games, and we are comparing both exponential and hyperbolic discounting to geometric discounting.

2.3 Dynamic Discounting

We define static discounting as a discount function or factor that remains fixed throughout the duration of learning and dynamic discounting as a discount function or factor that updates while learning. Our second modification to G_t is using a novel confidence-based approach and François-Lavet et al.'s proposed increasing discount factor. Our novel approach uses the preference for actions output by the agent model, $\pi_\theta(a_t|s_t)$, as seen in the equation 2. We treat this as the agent's confidence in the choice because it represent the action perceived to return the largest reward. Using this as the discount rate allows us to replicate confidence. As $\pi_\theta(a_t|s_t)$ becomes larger, it entails–based on the agent's experience–action, a_t, is the best in it's experience. As a result, the agent should be more "confident" and discount the choice less. This is similar to how humans discount as described by Mikhail et al. who studied how financial security forecasters become more accurate with their predictions as they gain more experience and react less to new information. The final G_t formulas used in our experiments are:

$$G_t = \sum_{i=0}^{N} r_i e^{-\pi_\theta(a_i|s_i)i} \tag{8}$$

and

$$G_t = \sum_{i=0}^{t} r_i (1 + \pi_\theta(a_i|s_i)i)^s \tag{9}$$

where $\pi_\theta(a_i|s_i)$ has replace γ, we use hyperbolic and exponential discounting, and the discount rate is now based on the agent's preference for the action. Now, it more closely resembles human discounting as it depends more on the agent's experience instead of a fixed hand-authored value as seen in most of the RL literature.

François-Lavet et al. implemented a DQN (Mnih et al., 2015) with an increasing discount factor to show that it takes less training time to converge, and when paired with a varying learning rate outperforms the baseline DQN. Their increasing discount rate is represented by the following equation:

$$\gamma_{k+1} = 1 - 0.98(1 - \gamma_k) \tag{10}$$

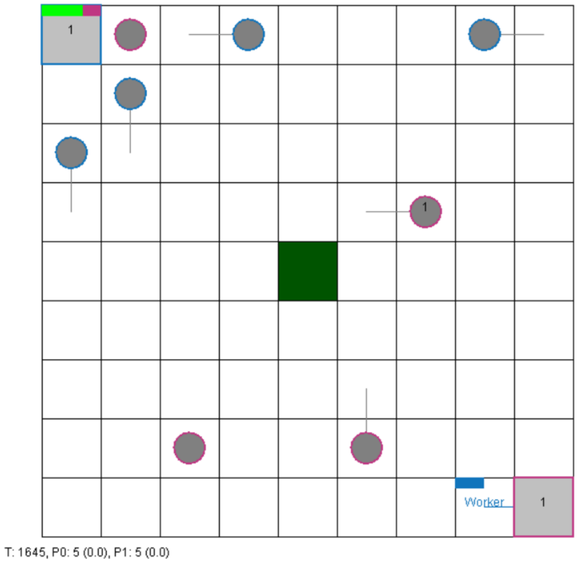
where γ_{k+1} is the increased discount rate (we increased ours 100k time steps) and γ_k is the current discount rate. This is different from the work discussed in this paper in that our domain has delayed rewards and immediate rewards and we use delay discounting functions while François-Lavet et al. uses games in the ALE domain that only have immediate rewards and they use a geometric discount function. Also, François-Lavet et al. uses a monotonically increasing discount factor, and ours changes with our model's predictions making it more human-like.

Fedus et al. implemented a meta-like discount factor for their ALE agent. For their ALE agent, they incorporated a multi-horizon auxiliary task where the agent used multiple discount rates/horizons for learning tasks with immediate rewards. They found this improved the performance of the agent over the baseline agent. As mentioned above, they also use hyperbolic discounting in one of their experiments. Our work is different with respect to hyperbolic discounting because we use a domain with delay rewards and use a dynamic discount factor with our hyperbolic discounting agent.

2.4 Time Consistent Discounting

An important property of discount functions is time consistency (Lattimore & Hutter, 2011; Shapiro & Pichler, 2016). Discount functions that are time consistent do not result in an agent changing its decision later. This property is important when searching for optimal behaviors because with a time consistent discount function the agent will never stray from the optimal policy. Therefore, the Bellman equation (Bellman, 1957) will be satisfied when using a time consistent discount function. Of the discount function that we utilize in our experiments, the exponential function and geometric function are time consistent. However, as mentioned above, these functions do not properly capture how humans discount. Studies have shown that hyperbolic discounting, which is considered time inconsistent, best match how humans discount choices.

Lattimore & Hutter and Shapiro & Pichler claim that it is important to avoid time inconsistent discount functions when possible. However, researchers have found that overconfidence is prevalent in complex domains where it is difficult to find an optimal policy (Klayman et al., 1999; Kausel et al., 2021). Changing preferences, based on newly gained information or could potentially prevent an agent from getting trapped in a bad policy unlike time consistent discount functions which avoid changing. This is why we believe it is important to find appropriate discount functions and learning behaviors based on the task or task difficulty because humans use these systems together to improve their behavior. Our confidence-based approach that we discuss later may also lead to time inconsistent behavior, but assuming the agent is following an optimal policy the agent should not change its mind.



T: 1645, P0: 5 (0.0), P1: 5 (0.0)

Figure 1. Example of the CTF map. Blue outline on units indicates player 2 and red outline indicates player 1. Squares indicates stationary units and circles indicate movable units. The green square is a resource stockpile that is treated as the flag. The light grey circles are the units that the agent controls to play.

3. Domain

3.1 Capture the Flag (CTF)

For our experimental domain, we recreated the game, Capture the Flag (CTF), in Gym-μRTS (Huang et al., 2021). It is an open real-time strategy game environment designed specifically for AI research (Ontañón et al., 2018). RTS games are two-player, zero-sum, simultaneous move games that simulate large army battles. The main challenges RTS games pose to players are huge decision spaces and actions have duration and can be issued simultaneously (Ontañón et al., 2018). For example, the branching factor (i.e. how fast the decision space grows) for Chess and Go has been estimated to be 35 and 180, respectively, while Gym-μRTS has a branching factor of 1022 (Ontañón et al., 2018). This makes RTS games a suitable domain for exploring discount functions because players have to consider not only a range of options but also the time and likelihood of completing their desired actions. Illustrating the importance and need of learning an effective discount function. Gym-μRTS simplifies this domain by reducing the types of units that the player can produce, but it still poses the same challenges as larger and more complex RTS games (e.g. StarCraft 2) and is configurable meaning more unit types, maps, and agents can be added.

We chose CTF as our domain because it is not as complex as a full RTS game, but still requires majority of the same sequences of actions (e.g. collecting resources, producing units, and attacking units) needed for learning to play full games. In our CTF implementation, we place one flag in the center of a 9x9 map and give each team enough resources to produce up to five worker units (the only units capable of harvesting resources or "picking up the flag"). The teams have 2000 time steps

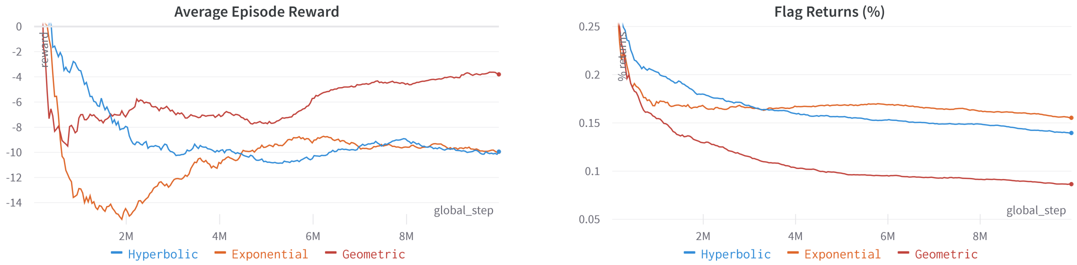


Figure 2. Average reward and flag return (%) for discount function

to capture and return the flag to their base. If a team is successfully able to return the flag to their base, they win the round. Otherwise, the round ends in a draw. Our reward structure/shaped reward gave +100 for winning, -100 for a draw/timeout, -100 for losing, +100 for return the flag, +1 for attacking the flag holder, and +0.1 for holding the flag. Rewards are given when the event is first initialized (Huang et al., 2021).

The challenge of capture the flag is the agent has to learn how to balance capturing the flag (pick up and return to base) and attacking (defend flag holder, attack flag holder, etc.). This poses a challenge for the agent because it is controlling up to five units at once, and has to issue commands to each to coordinate the units in order to achieve its goal. Depending on its discount function, the agent may prioritize sooner rewards (e.g. attacking enemies or holding the flag) over later rewards (returning flag) potentially leading to impulsive behavior and less success over time.

4. Delay Discounting

4.1 Study 1: Discount Functions

4.1.1 Performance

We found that all of the exponential and hyperbolic discounting agents have similar performance when we look at the average episode reward and flags captured. This suggests that both are suitable discount functions for this task, and improving the performance of agents further will rely on the information used for optimizing the agent. Geometric discounting, on the other hand, may have a similar episodic reward average but it is significantly less successful at learning to capture the flag. This suggests that the agents using the geometric function show signs of impulsiveness by focusing on immediate small rewards, instead of the larger reward that it receives from capturing the flag. We will discuss potential reasons for this behavior further in the next section. The difference in performance between the delay discounting functions and geometric function may be a result of how each function values time.

Setting	Attacks Issued	Units Produced	Flags Grabbed	Time Holding Flag
Exponential	19.56 ± 9.08	17.44 ± 5.49	3.28 ± 1.41	251.11 ± 273.7
Geometric	32.72 ± 10.35	27.78 ± 9.55	8 ± 6.98	545.06 ± 481.98
Hyperbolic	22.67 ± 9.57	19 ± 6.16	2.83 ± 1.76	209.83 ± 254.66

Table 2. Averages for actions with immediate rewards for each discount function

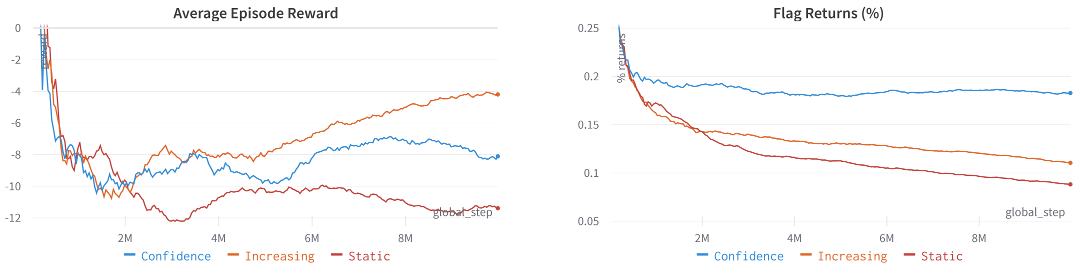


Figure 3. Average reward and flag return (%) for each discount rate

4.1.2 Behavior

For the agent's behavior we focus on the metrics: units produced, time holding flag, times the flag is grabbed, and enemy flag holders eliminated. Again we found that the hyperbolic and exponential agents behave similarly, with the hyperbolic agents spending a little more time holding the flag and eliminating less flag holders. This suggests both delay discounting functions create behaviors that exhibit self-control (i.e. capable of balancing immediate and long-term rewards). The geometric function has more impulsive behavior. As we can see in all four figures (attacking, units, flags grabbed, and time held), the geometric agents have significantly higher averages and each has a positive, increasing slope. This means they focused on accumulating more of the smaller rewards rather than attempting to capture the flag for a larger reward. Also, while visually observing the agents play, we noticed that they would often grab the flag then try to form a blockade around the flag holder. This behavior was not observed in the delay discounting agents.

4.2 Study 2: Static versus Dynamic Discounting

4.2.1 Performance

Focusing on just the episode reward and flags captured, we found that our confidence-based discount scheme has the lowest average episode reward but has the highest average flag captured. While the increasing and static discounting schemes both have higher average episode rewards, both have declining averages for flag captures. This behavior suggests that both schemes exhibit impulsiveness and prefer immediate rewards, while our confidence-based scheme appears to try to balance the immediate rewards and delayed rewards. We assume this difference in behavior is a result of overconfidence. Both the increasing and static schemes have large γ's (90+), meaning they always have

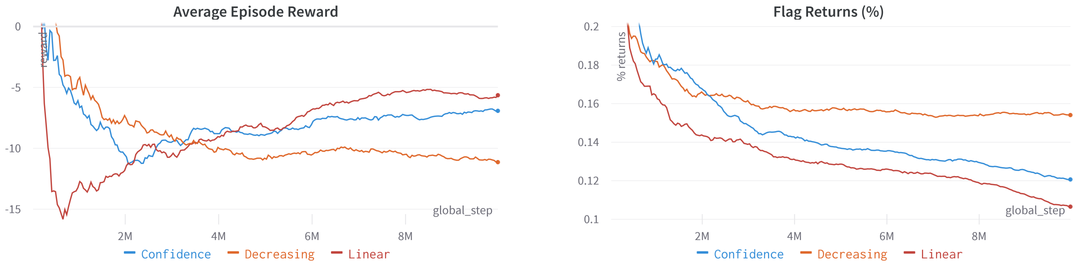


Figure 4. Average reward and flag return (%) for learning rates

a high belief of success in their actions regardless of their experience. This mindset in complex domains, as seen in the Ducati case study (Gino & Pisano, 2011) and discussed by Klayman et al. and Kausel et al., can lead to inaccuracies and misjudgements that are detrimental to performance.

4.2.2 Behavior

Setting	Attacks Issued	Units Produced	Flags Grabbed	Time Holding Flag
Confidence	25.72 ± 10.02	17.83 ± 5.63	3 ± 1.72	123.22 ± 117.43
Increasing	25.39 ± 9.84	23.67 ± 9.09	5.72 ± 6.51	441.28 ± 394.85
Static	23.83 ± 13.5	22.72 ± 9.48	5.39 ± 4.6	441.5 ± 449.08

Table 3. Averages for actions with immediate rewards for each discount rate

Reviewing the other metrics, we found the confidence agents have the most consistent behavior. They have significantly lower averages for three out of four of the metrics, and if we correlate this to its performance (average rewards and flags captured), it suggests the confidence agents focused more on delayed rewards. Both the increasing and static discount schemes behave similarly which we assume may be the result of overconfidence and prioritizing immediate returns. Since these four actions occur more frequently than capturing the flag, we believe the agents focus on accumulating smaller rewards instead of aiming for the larger, later reward. This is especially true since the agents have a constantly high or increasingly high level of "confidence" because it has a high, fixed discount rate or a monotonically increasing discount rate leading it to be believe that immediate rewards are better.

4.3 Study 3: Learning Rates

4.3.1 Performance

The decreasing learning rate proposed by François-Lavet et al. had the highest performance. We believe this is due to how quick the learning rate decreases. If the agent has success returning flags early enough in training, the decreasing learning rate was already low enough that it was underreacting to the new experiences. As a result, the agent would be more likely to maintain a successful policy. Our confidence-based learning rate also had some interesting results. The confidence-based learning rate shows a lot of gradual oscillate where the agent seemed to find a policy that was working (lowering the learning rate to "optimize") then later found that it stopped working (increasing the learning rate to find a better policy).

4.3.2 Behavior

Setting	Attacks Issued	Units Produced	Flags Grabbed	Time Holding Flag
Confidence	27.89 ± 11.63	22.89 ± 9.34	5.22 ± 6.04	323.67 ± 357.78
Decreasing	22.17 ± 8.94	17.67 ± 5.62	3.56 ± 1.89	224.94 ± 268.5
Linear	24.89 ± 12.2	23.67 ± 9.15	5.33 ± 5.39	457.39 ± 462.5

Table 4. Averages for actions with immediate rewards for each learning rate

Observing the behavior of these agents we can see that the quicker, decreasing learning rates performed better and resulted in less actions performed. This suggests that they were able to learn to return the flag early enough that they underreact to the smaller immediate rewards. We can see this looking at the Decreasing learning rate. It has double the flag captures of the other two learning rates and also has lower averages for all the actions for immediate rewards. The Linear learning rate on the other hand learns for longer and makes larger updates, and we believe this representative of overreactionary behavior. The agents may have some games where they do not capture the flag, and when they update their behavior they will put more emphasize on the immediate rewards that they were able to accumulate instead of making smaller changes to what they know is already "successful" for their task,

5. Conclusions and Future Work

In this paper, we explore discounting methods in reinforcement learning agents for explicitly modeling varying degrees of self-control in agents. Our findings suggest that a combination of discount functions, discount rate, and adaptive learning rate is necessary for modeling self-control in learning agents. Self-control is an important cognitive phenomena to computationally model due to its importance in human self-regulation issues of addiction, obesity, and other potentially harmful habits. We show through experiments in a real-time strategy game domain that we can produce behaviors using delay and dynamic discounting that representative of findings in behavioral psychology literature from human studies. We also show that the agents that use less impulsive discount rates and faster learning rates do not have the highest average reward but they complete the task with higher successful actions with longer-horizon tasks (flag returns in CTF).

Our future work will focus on more effectively incorporating delay discounting into reinforcement learning through anticipatory thinking and utilizing information theory to help our agents make more informed decisions. Anticipatory thinking, or more specifically episodic future thinking, is one's ability to simulate, or imagine, their personal future (Schacter et al., 2017). Current RL approaches, including our work in this paper, do not utilize anticipatory thinking. They discount the actual action and reward they receive assuming that they will survive long enough to receive the reward (Sutton & Barto, 2018). Based on our findings from this paper, we observe that it may be necessary to incorporate anticipatory thinking in order to fully model human-like decision making with delay discounting. Therefore, we will extend this work by exploring techniques to add episodic future thinking to our agents.

To complement the episodic future thinking work, we will explore how we can utilize information theory to provide more information to our learning agents. Information theory describes how well a statistical model captures a random variable (Cover, 1999). To achieve this we will shift towards a distributional RL approach which will provide us with a distributional approximation of the reward function instead of a single, scalar value like standard RL approaches (Bellemare et al., 2017). Combining a distributional RL approach with information theory will enable us to start accounting for risks and uncertainty in our agent's decisions. This will enable us to better utilize the delay discounting methods when paired with episodic future thinking because we can calculate subjective values for decisions.

6. Acknowledgements

We thank the microRTS development team for providing us with the base code for creating the Capture-the-flag game. Costa Huang and Santi Ontañón for insightful discussions. We thank anonymous reviewers for their comments on the paper and suggestions for improvement.

References

Bellemare, M. G., Dabney, W., & Munos, R. (2017). A distributional perspective on reinforcement learning. *International Conference on Machine Learning* (pp. 449–458). PMLR.

Bellman, R. (1957). A markovian decision process. *Journal of mathematics and mechanics*, *6*, 679–684.

Cover, T. M. (1999). *Elements of information theory*. John Wiley & Sons.

Fedus, W., Gelada, C., Bengio, Y., Bellemare, M. G., & Larochelle, H. (2019). Hyperbolic discounting and learning over multiple horizons. *arXiv preprint arXiv:1902.06865*.

François-Lavet, V., Fonteneau, R., & Ernst, D. (2015). How to discount deep reinforcement learning: Towards new dynamic strategies. *arXiv preprint arXiv:1512.02011*.

Gino, F., & Pisano, G. P. (2011). Why leaders don't learn from success. *Harvard business review*, *89*, 68–74.

Green, L., & Myerson, J. (2004). A discounting framework for choice with delayed and probabilistic rewards. *Psychological bulletin*, *130*, 769.

Huang, S., Ontañón, S., Bamford, C., & Grela, L. (2021). Gym-μ rts: Toward affordable full game real-time strategy games research with deep reinforcement learning. *arXiv preprint arXiv:2105.13807*.

Jessup, R. K., Bishara, A. J., & Busemeyer, J. R. (2008). Feedback produces divergence from prospect theory in descriptive choice. *Psychological Science*, *19*, 1015–1022.

Kahneman, D., & Tversky, A. (2013). Prospect theory: An analysis of decision under risk. In *Handbook of the fundamentals of financial decision making: Part i*, 99–127. World Scientific.

Kausel, E. E., Carrasco, F., Reyes, T., Hirmas, A., & Rodríguez, A. (2021). Dynamic overconfidence: a growth curve and cross lagged analysis of accuracy, confidence, overestimation and their relations. *Thinking & Reasoning*, *27*, 417–444.

Klayman, J., Soll, J. B., Gonzalez-Vallejo, C., & Barlas, S. (1999). Overconfidence: It depends on how, what, and whom you ask. *Organizational behavior and human decision processes*, *79*, 216–247.

Lattimore, T., & Hutter, M. (2011). Time consistent discounting. *arXiv preprint arXiv:1107.5528*.

Mikhail, M. B., Walther, B. R., & Willis, R. H. (2003). The effect of experience on security analyst underreaction. *Journal of Accounting and Economics*, *35*, 101–116.

Mischel, W. (1974). Processes in delay of gratification. In *Advances in experimental social psychology*, volume 7, 249–292. Elsevier.

Mnih, V., et al. (2015). Human-level control through deep reinforcement learning. *nature*, *518*, 529–533.

Odum, A. L. (2011). Delay discounting: I'm ak, you're ak. *Journal of the experimental analysis of behavior*, *96*, 427–439.

Ontañón, S., Barriga, N. A., Silva, C. R., Moraes, R. O., & Lelis, L. H. (2018). The first microrts artificial intelligence competition. *AI Magazine*, *39*, 75–83.

Schacter, D. L., Benoit, R. G., & Szpunar, K. K. (2017). Episodic future thinking: Mechanisms and functions. *Current opinion in behavioral sciences*, *17*, 41–50.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shapiro, A., & Pichler, A. (2016). Time and dynamic consistency of risk averse stochastic programs. *Optimization Online*.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Tversky, A., & Kahneman, D. (1992). Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and uncertainty*, *5*, 297–323.