Generalized Hyperbolic Discounting for Delay-Sensitive Reinforcement Learning

Raja Farrukh Ali
Kansas State University
rfali@ksu.edu

John Woods
Kansas State University
jwoods03@ksu.edu

Robert A Southern
Kansas State University
rsouthern@ksu.edu

Travis Smith
Kansas State University
smithtr@ksu.edu

Vahid Behzadan
University of New Haven
vbehzadan@newhaven.edu

William Hsu
Kansas State University
bhsu@ksu.edu

Abstract

Value estimates at multiple timescales can help create advanced discounting functions and allow agents to form more effective predictive models of their environment. While exponential discounting has been widely used because of its time-consistent preferences and ease of use, hyperbolic discounting has been shown to capture human and animal preferences more accurately. Both the exponential and hyperbolic reward discounting functions are single-parameter models. However, more sophisticated, two-parameter hyperbolic discounting functions have been proposed that provide the best fit to observed human behavior. In this work, we propose a generalized hyperbolic discounting framework, incorporating both a discount factor and a sensitivity-to-delay parameter through which agents have different valuations of the same time delay it takes to receive a reward. We conduct extensive evaluations across a variety of learning tasks (high dimensional input, generalization), analyze the suitability of different discounting functions to these tasks, and present new insights on how the functional form of discounting affects an agent's performance.

1 Introduction

In reinforcement learning (RL), rewards are typically discounted exponentially, meaning that a reward obtained t time steps in the future is discounted by a factor of γ^t (Bellman, 1957b; Sutton & Barto, 1998). This approach establishes a fixed learning horizon for the agent: a smaller γ value prioritizes short-term rewards, while a larger γ value emphasizes long-term rewards. However, human and animal behavior often follows hyperbolic discounting patterns (Mazur, 1987), characterized by the hyperbolic function $\frac{1}{1+kt}$, where $k > 0$ represents the hyperbolic discounting rate. Unlike exponential models, hyperbolic discounting accounts for preference reversal over time (Green et al., 1994) and offers better alignment with decision-making scenarios involving multiple reward variables, such as delay length, reward magnitude, and probability (Green & Myerson, 2004).

In addition to discounting rewards that are received after a delay, individual differences in the perception of delay also exist. While one individual may exhibit impatience by preferring immediate rewards, another may display patience by choosing to wait for the delayed reward. Human studies have demonstrated that the single-parameter hyperbolic (base model) tends to overestimate the perceived value of shorter delays and underestimate the value of longer delays (McKerchar et al., 2009). To address these individual variations, a sensitivity-to-delay parameter, denoted as s where $0 < s < 1$, has been introduced as a second parameter to the hyperbolic function. Two methods have been proposed for this purpose: one by Rachlin (1989), which applies s solely to the delay term, yielding $\frac{1}{1+kt^s}$, and another by Myerson & Green (1995), which applies s to the entire denominator

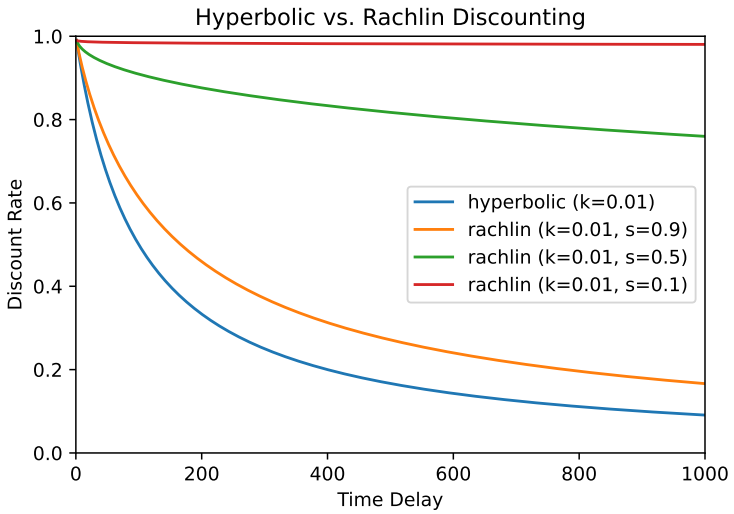


Figure 1: Single-parameter Hyperbolic Discounting vs Two-parameter Generalized Hyperbolic (also known as Rachlin, after its inventor (Rachlin, 1989)). The Rachlin hyperbolic discounting allows variation not only in the rate of discounting (k), but also sensitivity to delay (s). A lower value of s indicates less sensitivity to delay, meaning that the subjective value of a reward decreases less rapidly as the delay increases. Conversely, a value closer to 1 indicates a higher sensitivity to delay, with the subjective value decreasing more rapidly.

resulting in $\frac{1}{(1+kt)^s}$. When fixed at $s = 1$, both models simplify to Mazur's one-parameter hyperbolic discounting model ($\frac{1}{(1+kt)}$).

To illustrate the differences, let us consider an example. Suppose you are given the choice to receive \$50 today or \$100 one month from now. Considering the immediate benefit of choosing \$50 now versus waiting much longer for \$100, many would choose the \$50 option. Now suppose you are given the choice to receive \$50 in twelve months or \$100 in thirteen months. The exponential discounting model predicts you would still choose \$50, since the \$100 option is again delayed by an extra month so the proportion discounted remains the same. However, many people would choose \$100 in this scenario, since the difference between 12 and 13 months is relatively smaller than the difference between 0 and 1 month. Sensitivity-to-delay models come into play if your answers differ from the above. If you were willing to wait a month in the first scenario and chose \$100, then you exhibit lower sensitivity to delays (s closer to 0). If you choose not to wait a month in the second scenario and choose \$50, then you have higher sensitivities toward delays (s closer to 1). This is a prime example demonstrating preference reversal, which is properly accounted for by hyperbolic discounting models, and how sensitivity-to-delay models can accommodate individual preferences.

The Rachlin model provides certain advantages over the base model (Franck et al., 2023); it offers flexibility and aligns closely with empirical discounting data. Moreover, compared to other two-parameter discounting models, the Rachlin model provides the advantage of easily obtaining unique best estimates for parameters across a wide range of potential discounting patterns. Figure 1 depicts the one-parameter hyperbolic discounting and two-parameter Rachlin model of hyperbolic discounting, with three different values of the sensitivity-to-delay parameter s shown. The main contribution of this work is to demonstrate a practical way of integrating the two-parameter Rachlin model into deep RL to evaluate whether RL agents respond differently to changes in delay sensitivity while examining the potential benefits and drawbacks of such a technique. We examine Rachlin discounting in the off-policy value-based RL algorithm Rainbow (Hessel et al., 2018) and evaluate the performance of the Rachlin model for three different values of the sensitivity-to-delay parameter s on a variety of RL benchmarks such as Atari-5 (Aitchison et al., 2023) and Procgen (Cobbe et al., 2020). Our results

show that the applicability of sensitive-to-delay discounting varies greatly with the environment properties. In more deterministic environments like Atari-5, the agent has no incentive to decrease in perceived delay. However, for environments like Procgen that are used to study generalization, there is a consistent performance improvement as the agent becomes less sensitive to perceived delay (s value closer to 0).

2 Related Work

Temporal discounting, a key concept in economics and decision theory, posits that individuals value immediate rewards more than delayed ones. Traditional models use exponential discounting, where reward value decreases at a constant rate over time (Samuelson, 1937). However, empirical studies show human preferences align better with hyperbolic discounting, which captures steeper discounting for shorter delays (Ainslie, 1975; Mazur, 1987). Research in psychology and behavioral economics has extensively explored these preferences, revealing that hyperbolic models more accurately describe human behavior, including preference reversals (McKerchar et al., 2009; Smith et al., 2023; Young, 2017). Sensitivity to delay models, like the Rachlin hyperboloid model, introduce a second parameter to better fit human data, accounting for variance more effectively than exponential or single-parameter models (Rachlin, 1989; McKerchar et al., 2009; Crystal, 2001; Myerson & Green, 1995). These models have significantly influenced studies on intertemporal choice, showing superiority over exponential discounting in explaining human decision-making (Green et al., 1994; Kirby, 1997).

Traditional RL algorithms, including Q-Learning and policy gradient methods, typically use exponential discounting to manage delayed rewards (Sutton & Barto, 1998). Kurth-Nelson & Redish (2009) modeled hyperbolic discounting via distributed exponential discounting, and Fedus et al. (2019) extended this to deep RL, approximating hyperbolic discounting through multi-horizon learning. Further advancements include meta-learning approaches to use γ as a learnable parameter (Xu et al., 2018), Γ-nets for value prediction across discount factors (Sherstan et al., 2020), non-exponential discounting for model-based RL (Schultheis et al., 2022), and beta-weighted discounting (Kwiatkowski et al., 2023), whereas another body of work has advocated for state dependent discounting (White, 2017; Pitis, 2019; 2023).

3 Methodology

3.1 Approximating Rachlin Q-values

In order to use Rachlin hyperbolic discounting in deep RL, we need to incorporate such a discounting function into a learning algorithm. Following Fedus et al. (2019), we demonstrate a method to re-purpose exponentially-discounted Q-values to compute Rachlin hyperbolic discounted Q-values. The Bellman equation (Bellman, 1957a) is given by

$$Q_\pi^{\gamma^t}(s,a) = \mathbb{E}_{\pi,P}[R(s,a) + \gamma Q_\pi(s',a')] \tag{1}$$

where the expectation $\mathbb{E}_{\pi,P}$ involves sampling $a \sim \pi(\cdot|s)$, $s' \sim P(\cdot|s,a)$, and $a' \sim \pi(\cdot|s')$. Let's consider estimating the value function under hyperbolic discounting. We denote Rachlin hyperbolic Q-values as $Q_\pi^{\Gamma_{k\sigma}}$, using σ instead of s to denote the sensitivity-to-delay parameter in this section to avoid confusion with the state s, as shown in Equation 3:

$$Q_\pi^{\Gamma_{k\sigma}}(s,a) = \mathbb{E}_\pi\left[\Gamma_{k\sigma}(1)R(s_1,a_1) + \Gamma_{k\sigma}(2)R(s_2,a_2) + \cdots \middle| s,a\right] \tag{2}$$

$$= \mathbb{E}_\pi\left[\sum_t \Gamma_{k\sigma}(t)R(s_t,a_t) \middle| s,a\right] \tag{3}$$

Remember that the value of a reward r_t at timestep t with hyperbolic exponent k and sensitivity-to-delay σ is give by:

$$V(t) = \frac{r_t}{1 + kt^\sigma} \tag{4}$$

We now relate the hyperbolic Q-value, Q^Γ_π, to exponential Q-value, Q^γ_π, learned through standard Q-learning. The hyperbolic discount $\Gamma_{k\sigma}$ can be expressed as the integral of a specific function $f(\gamma, t)$ for $\gamma = [0, 1)$ and $\sigma = [0, 1]$:

$$\int_0^1 \gamma^{kt^\sigma} d\gamma = \frac{1}{1 + kt^\sigma} = \Gamma_{k\sigma}(t) \tag{5}$$

This integral over the function $f(\gamma, t) = \gamma^{kt^\sigma}$ yields the desired hyperbolic discount factor $\Gamma_k(t)$ by considering an *infinite set* of exponential discount factors γ over its domain $\gamma \in [0, 1)$. Recognizing that the integrand γ^{kt} is the standard exponential discount factor suggests a connection to standard Q-learning. This implies that by considering an infinite set of γ, we can combine them to yield hyperbolic discounts for the corresponding time-step t. We employ Equation 5 to compute the $Q^{\Gamma_{k\sigma}}_\pi$-value according to the hyperbolic discount factor by considering an infinite set of $Q^{\gamma^k}_\pi$-values computed through standard Q-learning, as shown in Equation 9.

$$\begin{aligned}
Q^{\Gamma_{k\sigma}}_\pi(s, a) =& \mathbb{E}_\pi \left[\sum_t \Gamma_{k\sigma}(t) R(s_t, a_t) \middle| s, a \right] && (6)\\
=& \mathbb{E}_\pi \left[\sum_t \left(\int_{\gamma=0}^1 \gamma^{kt^\sigma} d\gamma \right) R(s_t, a_t) \middle| s, a \right] && (7)\\
=& \int_{\gamma=0}^1 \mathbb{E}_\pi \left[\sum_t R(s_t, a_t) (\gamma^{kt})^\sigma \middle| s, a \right] d\gamma && (8)\\
=& \int_{\gamma=0}^1 Q^{(\gamma^{kt})^\sigma}_\pi (s, a) d\gamma && (9)
\end{aligned}$$

This approach demonstrates how to compute hyperbolic Q-values by considering an infinite set of exponential Q-values, each corresponding to a different discount factor γ. The number of concurrent horizons (n_γ) is an important factor to consider, along with the values of γ that enforce the minimum and maximum horizon for the agent, beyond which the rewards are negligible. In practice, we start by calculating the γ interval, which are the values of γ on which the integral is approximated using a Riemann sum, and are specified by γ^k. The γ^k value thus obtained is raised to the power of t^σ, and the factor γ^{kt^σ} can be considered as the Bellman gamma, the value of gamma that is used for learning in Q-Learning (Eq. 1). Note that Rachlin hyperbolic discounting necessitates the use of **n-step** temporal difference learning, as traditional 1-step learning with $t = 1$, renders the sensitivity to delay parameter σ ineffective, i.e. $t^\sigma = 1$.

3.2 Model Architecture

We evaluate Rachlin discounting in value-based model-free RL by using Rainbow (Hessel et al., 2018) as our base algorithm. The proposed network architecture is presented in Figure 2. We employ the deeper IMPALA-CNN architecture (Espeholt et al., 2018) with 15 convolutional layers instead of the small 3-layer network (Nature-CNN) used in Fedus et al. (2019). The residual blocks (He et al., 2016) in the IMPALA-CNN architecture keep the optimization process light while enabling substantially deeper feature learning. The network predicts Q-values for each discount factor γ, and these Q-values are used for the agent's learning (loss calculation). Rainbow combines several

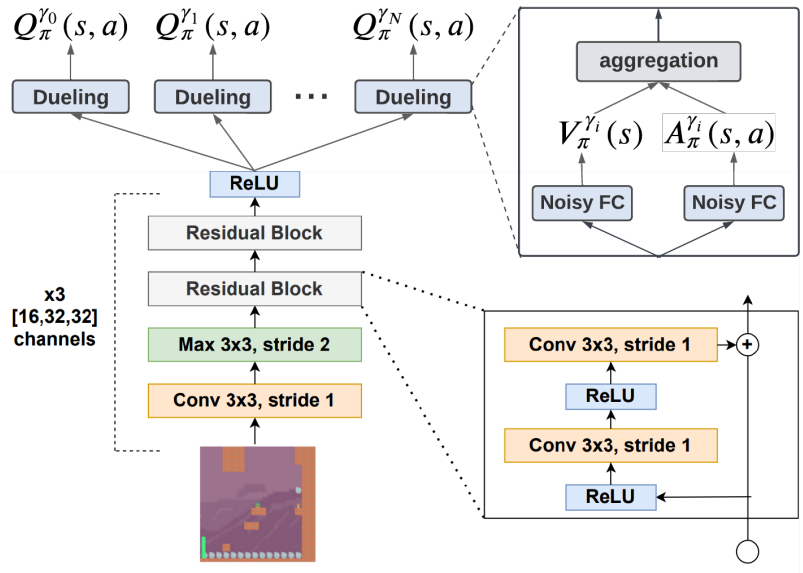


Figure 2: Network architecture. Output layers predict Q-values for different discount factors using an individual output block for each γ, which are then used to approximate the Hyperbolic Q-value.

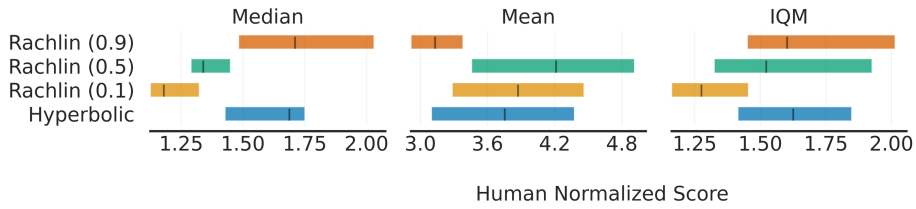
independent improvements on top of the Deep Q-Learning framework (Mnih et al., 2015). We include five of the six; double DQN (van Hasselt et al., 2016), dueling DQN (Wang et al., 2016), noisy nets (Fortunato et al., 2018), prioritized experience replay buffer (Schaul et al., 2016), and n-step returns (Sutton, 1988). However, we exclude the Distributional RL (C51) (Bellemare et al., 2017) component as we trade off implementation complexity with performance benefits, particularly by evaluating for a lesser number of time steps (25M). As noted by Hessel et al. (2018), optimizing the distribution of returns helps in the long run, such as training beyond 40M time steps. Schmidt & Schmied (2021) also note marginal performance improvement with the inclusion of C51 when training for limited time steps (10M) and suggest the exclusion of this component.

4 Experiments

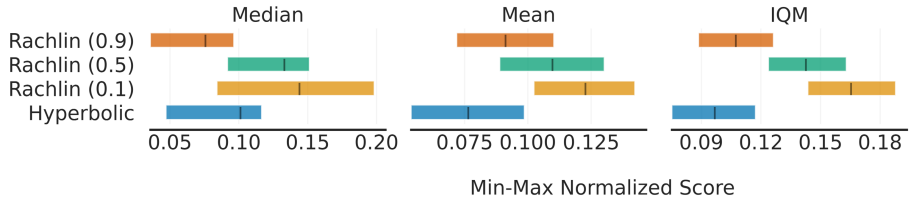
We evaluate our proposed approach on a suite of tasks designed to test the agent's ability to make decisions involving intertemporal trade-offs. These tasks include delayed gratification scenarios, where the agent must choose between an immediate smaller reward or a larger delayed reward, as well as more complex environments that require long-term planning and decision-making. We conducted experiments to assess the effectiveness of Rachlin hyperbolic discounting across two benchmark environments: Atari-5 (Aitchison et al., 2023) and Procgen (Cobbe et al., 2020). For all experiments, the hyperbolic discount factor $k = 0.1$ remained constant. We varied the n-step values between 3 and 20 (results are only shown for $n_{step} = 3$) and compared the performance of Rachlin discounting across three arbitrary selections of $s \in \{0.1, 0.5, 0.9\}$. We compared with single-parameter hyperbolic discounting as a baseline. We do not perform hyperparameter tuning. All agents are evaluated using undiscounted, episodic returns, following the guidelines of Agarwal et al. (2021). Experimental setup and hyperparameter details are provided in Appendix A.

4.1 Atari-5

We first conduct experiments on Atari-5, a subset of 5 environments from the Arcade Learning Environment (ALE) (Bellemare et al., 2013) as presented by Aitchison et al. (2023), which produces 57-game median score estimates within 10% of their true values. The aggregated results (Figure 3a) show that in ALE, higher values of σ fare better, with hyperbolic discounting ($\sigma = 1$) achieving



(a) Atari-5



(b) Procgen

Figure 3: Aggregated results for the Generalized Hyperbolic discounting function (Rachlin, 1989) and the single-parameter Hyperbolic Discounting function (Mazur, 1987) across the Atari-5 and Procgen benchmarks. Results were aggregated across all constituent environments (5 in Atari-5, 16 in Procgen) and each environment/method combination was run for 5 seeds.

the best overall IQM score. This implies that sensitivity to delay (smaller σ values) is penalized for agents, and difference between actual delay and perceived delay needs to be minimal. Full results shown in Appendix B, Figure 4.

4.2 Procgen

Procgen consists of 16 unique environments which are designed to assess an agent's generalization ability by evaluating agents on levels it has not encountered during training. Figure 3b shows the performance of Rachlin hyperbolic discounting for 3 different values of s against the one-parameter Hyperbolic discounting (baseline). Surprisingly, the results for Procgen are opposite of ALE, and lower values of σ (more sensitive to delay) increase the agent's overall performance. Full results shown in Appendix B, Figure 5.

5 Discussion

The conventional exponential reward discounting model in reinforcement learning fails to capture the intricacies of human decision-making processes. Individual subjective value of a reward varies based on the perceived time required to obtain it. In this work, we introduced a novel approach to integrating the Rachlin hyperbolic discounting model into the deep reinforcement learning framework. By modifying the value estimation process to incorporate both the discount factor and the sensitivity-to-delay parameter, our approach enables RL agents to learn policies that align better with observed human preferences and decision-making patterns in tasks involving intertemporal trade-offs. Our results demonstrate the potential of this approach to improve decision-making and performance in delayed gratification scenarios and other tasks requiring generalizability to unseen tasks. Our results show that the optimal value of σ varies across different environments, indicating that environmental dynamics significantly influence this parameter. Fine-tuning σ during hyperparameter optimization may enhance the performance of Rachlin discounting.

By integrating sensitivity-to-delay models from psychological sciences into reinforcement learning agents, we have demonstrated a method to incorporate subjective value and accommodate preference reversals. This approach holds promise for developing AI agents that emulate human behavior more accurately, enhancing social acceptability and facilitating smoother collaboration with humans.

Future work could explore integrating our approach into more complex RL algorithms, such as multi-agent reinforcement learning or hierarchical reinforcement learning frameworks. Further investigation into the theoretical properties and convergence guarantees of our approach would be valuable, as well as exploring alternative formulations of the two-parameter delay discounting function.

References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *NeurIPS*, 2021.

George Ainslie. Specious reward: A behavioral theory of impulsiveness and impulse control. *Psychological Bulletin*, 82(4):463–496, 1975. doi: 10.1037/h0076860.

Matthew Aitchison, Penny Sweetser, and Marcus Hutter. Atari-5: Distilling the arcade learning environment down to five games. In *International Conference on Machine Learning*, pp. 421–438. PMLR, 2023.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pp. 449–458. PMLR, 2017.

Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1957a.

Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, pp. 679–684, 1957b.

Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *ICML*, pp. 2048–2056. PMLR, 2020.

Jonathon D Crystal. Nonlinear time perception. *Behavioural Processes*, 55(1):35–49, 2001. ISSN 0376-6357. doi: https://doi.org/10.1016/S0376-6357(01)00167-X. URL https://www.sciencedirect.com/science/article/pii/S037663570100167X.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International Conference on Machine Learning*, pp. 1407–1416. PMLR, 2018.

William Fedus, Carles Gelada, Yoshua Bengio, Marc G Bellemare, and Hugo Larochelle. Hyperbolic discounting and learning over multiple horizons. *arXiv preprint arXiv:1902.06865*, 2019.

Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Ian Osband, Alex Graves, Vlad Mnih, Remi Munos, Demis Hassabis, Olivier Pietquin, et al. Noisy networks for exploration. *Proceedings of the International Conference on Representation Learning (ICLR)*, 2018.

Christopher T Franck, Haily K Traxler, Brent A Kaplan, Mikhail N Koffarnus, and Mark J Rzeszutek. A tribute to howard rachlin and his two-parameter discounting model: Reliable and flexible model fitting. *Journal of the Experimental Analysis of Behavior*, 119(1):156–168, 2023.

Leonard Green and Joel Myerson. A discounting framework for choice with delayed and probabilistic rewards. *Psychological bulletin*, 130(5):769, 2004.

Leonard Green, Nathanael Fristoe, and Joel Myerson. Temporal discounting and preference reversals in choice between delayed outcomes. *Psychonomic Bulletin & Review*, 1(3):383–389, 1994.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *AAAI conference on artificial intelligence*, 2018.

Kris N. Kirby. Bidding on the future: Evidence against normative discounting of delayed rewards. *Journal of Experimental Psychology: General*, 126(1):54–70, 1997.

Zeb Kurth-Nelson and A David Redish. Temporal-difference reinforcement learning with distributed representations. *PLoS One*, 4(10):e7362, 2009.

Ariel Kwiatkowski, Vicky Kalogeiton, Julien Pettré, and Marie-Paule Cani. Ugae: A novel approach to non-exponential discounting. *arXiv preprint arXiv:2302.05740*, 2023.

James E. Mazur. An adjusting procedure for studying delayed reinforcement. In *Quantitative analyses of behavior*, volume 5, pp. 55–73. Lawrence Erlbaum Associates, Inc., 1987.

Todd L McKerchar, Leonard Green, Joel Myerson, T Stephen Pickford, Jade C Hill, and Steven C Stout. A comparison of four models of delay discounting in humans. *Behavioural processes*, 81(2): 256–259, 2009.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Joel Myerson and Leonard Green. Discounting of delayed rewards: Models of individual choice. *Journal of the experimental analysis of behavior*, 64(3):263–276, 1995.

Silviu Pitis. Rethinking the discount factor in reinforcement learning: A decision theoretic approach. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 7949–7956, 2019.

Silviu Pitis. Consistent aggregation of objectives with diverse time preferences requires non-markovian rewards. *Advances in Neural Information Processing Systems*, 36, 2023.

Howard Rachlin. *Judgment, Decision, and Choice: A Cognitive/Behavioral Synthesis*. W.H. Freeman and Company, 1989.

Paul A Samuelson. A note on measurement of utility. *The review of economic studies*, 4(2):155–161, 1937.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *ICLR*, 2016.

Dominik Schmidt and Thomas Schmied. Fast and data-efficient training of rainbow: an experimental study on atari. *Deep RL Workshop NeurIPS*, 2021. URL https://arxiv.org/abs/2111.10247.

Matthias Schultheis, Constantin A Rothkopf, and Heinz Koeppl. Reinforcement learning with non-exponential discounting. *Advances in neural information processing systems*, 35:3649–3662, 2022.

Craig Sherstan, Shibhansh Dohare, James MacGlashan, Johannes Günther, and Patrick M Pilarski. Gamma-nets: Generalizing value estimation over timescale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5717–5725, 2020.

Travis R Smith, Robert Southern, and Kimberly Kirkpatrick. Mechanisms of impulsive choice: Experiments to explore and models to map the empirical terrain. *Learning & behavior*, 51(4): 355–391, 2023.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3 (1):9–44, 1988.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT press Cambridge, 1998.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*. PMLR, 2016.

Martha White. Unifying task specification in reinforcement learning. In *International Conference on Machine Learning*, pp. 3742–3750. PMLR, 2017.

Zhongwen Xu, Hado P van Hasselt, and David Silver. Meta-gradient reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

Michael E Young. Discounting: A practical guide to multilevel analysis of indifference data. *Journal of the Experimental Analysis of Behavior*, 108(1):97–112, 2017.

Appendix

A Setup and Hyperparameters

The software used included Python (3.8.x) and PyTorch (1.12), with CUDA (11.4). Our computing infrastructure involved using 2 GPU servers, each having an Intel Xeon 4215 processor (3.50GHz), 16 physical cores, 1 TB RAM, and 8 Nvidia A40 GPUs. We provide the hyperparameters used in our implementation in Table 1. All figures in this paper are plotted using an exponential moving average with a smoothing value of 0.9, where each point on the graph is calculated by taking the mean of the last 5 observations (corresponding to the last 1M steps). In all plots, mean (dark line) and 95% bootstrapped confidence intervals (shaded region) are calculated over 5 runs using a different seed.

Hyperparameters	**Values**
Training protocol	25M steps
Evaluation protocol	10 episodes (every 250k train steps)
Max steps/episode	27k
Double DQN	Yes
Dueling DQN	Yes
σ_0 for Noisy Layers	0.5
Number of Atoms	Not Used
PER Importance sampling β_0	0.4
n-step	3
Learning rate	0.00025
Batch size	256
Q-target update frequency	8000 steps
Adam ϵ	2e-5 (0.005/batch size)
Parallel environments	64
Replay buffer size	$\approx$1M (2^{20})
Training starts at	20000 steps
Number of γ	5
γ_{max}	0.99
Hyperbolic exponent k	0.1
Integral estimate	lower
γ intervals	$[0.374, 0.608, 0.755, 0.847, 0.904]$
γ values	$[0.906, 0.951, 0.972, 0.985, 0.99]$
σ values	$[0.1, 0.5, 0.9]$
Procgen distribution mode	Easy
Procgen num levels (training)	200
Procgen num levels (evaluation)	0 (infinite)
Procgen start level	0
ALE Sticky Actions probability	0
ALE Frame Stacking	4
ALE Frame Skip	4
Number of seeds per environment	5
Random seed values	$[64331, 74330, 95762, 282995, 801604]$

Table 1: Hyperparameters for Rainbow

B Individual Results

B.1 Atari-5

Results on Atari-5 Aitchison et al. (2023), shown in Figure 4, confirm that the sensitivity-to-delay parameter affects learning, with the base hyperbolic performing best on only 1 of the 5 environments. We also note that different environments behave differently to the sensitivity-to-delay parameter, meaning that this hyperparameter may need to be carefully tuned before it can be applied. In our work, we did not do any hyperparameter tuning and selected 3 representative values from the range $[0, 1]$.

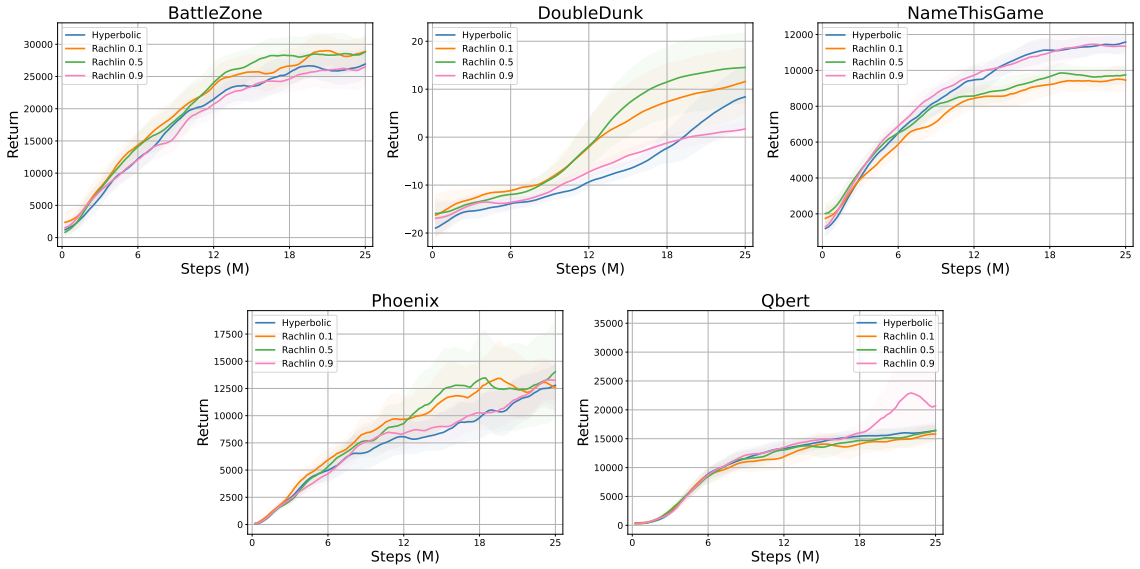


Figure 4: Performance of Rachlin hyperbolic discounting for 3 different values of sensitivity-to-delay (σ) on the Atari-5 benchmark (Aitchison et al., 2023). Mean (dark line) and 95% bootstrapped confidence intervals (shaded region) are shown, calculated over 5 seeds for each experiment.

B.2 Procgen

Figure 5 shows the performance of Rachlin hyperbolic discounting for 3 different values of σ against the one-parameter Hyperbolic discounting (baseline) on all Procgen environments for 25M timesteps. The performance of the baseline one-parameter hyperbolic is close to the 3 Rachlin models studied here, except in bigfish, coinrun and starpilot environments.

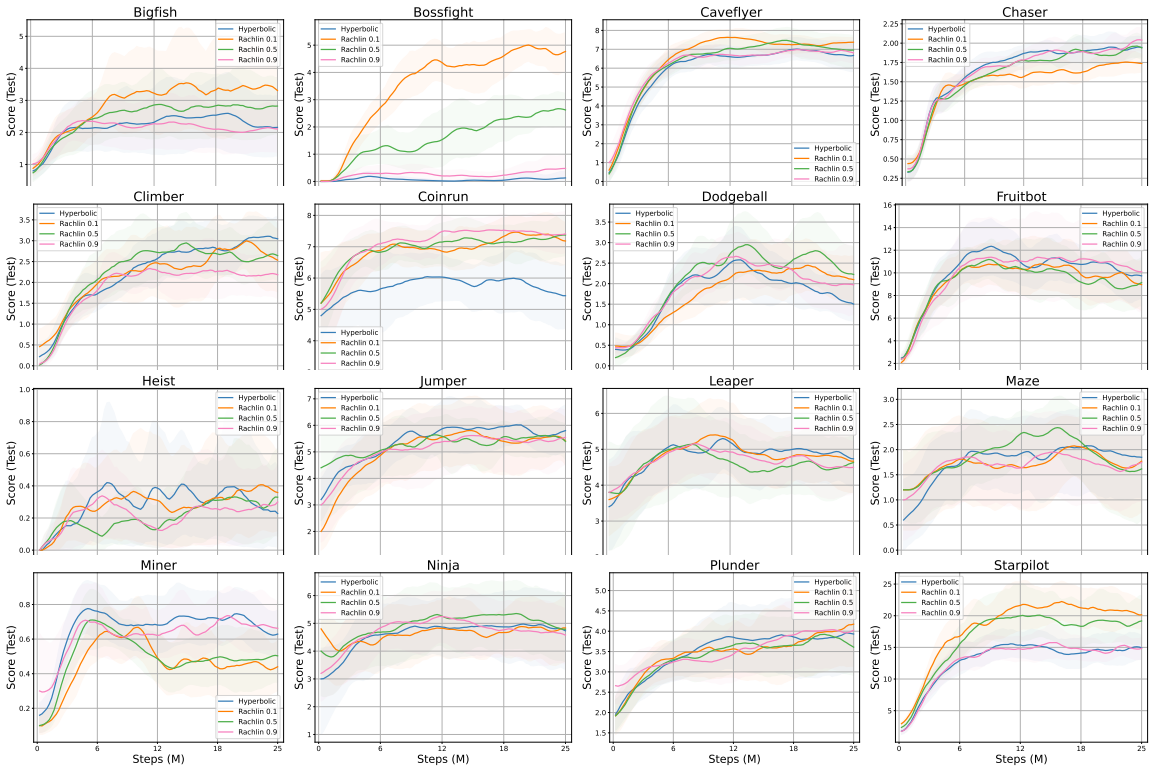
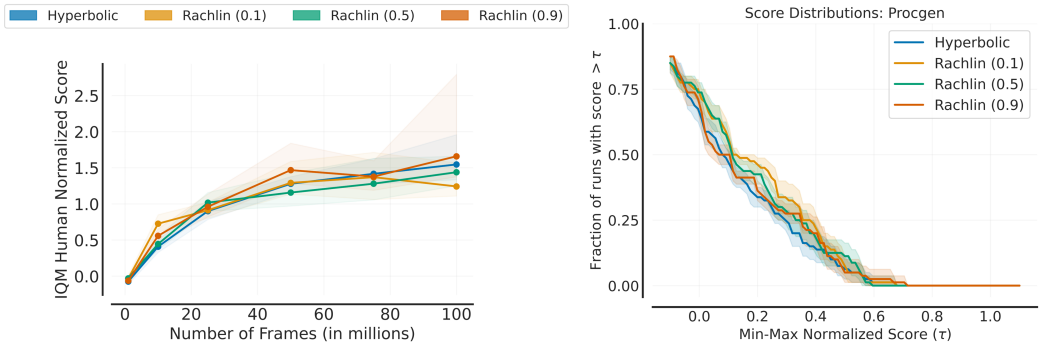


Figure 5: Performance of Rachlin hyperbolic discounting for 3 different values of sensitivity-to-delay (s) against the one-parameter Hyperbolic discounting (baseline) on the Procgen benchmark. Mean and 95% confidence intervals are shown.



(a) Atari-5 (b) Procgen

Figure 6: (a) Sample Efficiency curve for IQM Human-normalized score across Atari-5 tasks (b) Performance profile for Procgen tasks.