# An Empirical and Conceptual Categorization of Value-based Exploration Methods

Niko Yasui [1] Sungsu Lim [1] Cam Linke [1] Adam White [1 2] Martha White [1]

## Abstract

This paper categorizes several exploration methods for reinforcement learning according to their underlying heuristics. Using Q-learning with linear function approximation, we compare representative methods from each category on a set of domains designed to pose different exploratory challenges. We find that the relative performance of each method depends on the specific exploratory challenge posed by the domain. Our results suggest that each exploration heuristic encodes a bias which is appropriate for a subset of environments.

## 1. Introduction

Effective exploration strategies are critical for obtaining the data needed to learn optimal and near-optimal policies. Exploration has been well-studied in the literature, particularly for approaches that use state-transition models in the value-function update (see Szita & Lőrincz, 2008), called planning. Much of the work has been theoretical, and despite the large body of literature most control papers use simple strategies like $\epsilon$-greedy. While the planning-free setting has historically been studied less, difficulties around planning with function approximation have motivated the development of several new planning-free approaches. In this work we aim to clarify key ideas surrounding exploration with planning-free, value-based methods.

Planning-free exploration methods have largely been investigated in isolation, often compared only to simple baselines like $\epsilon$-greedy. This separation has made it difficult to gauge the relative performance of different strategies or identify shared concepts underlying the algorithms. Some previous work has categorized methods based on whether they direct exploration or explore randomly (Thrun, 1992) or based on subcategories of optimism in the face of uncertainty (OFU), including separating count-based and confidence-based methods (Kumaraswamy et al., 2018). Other studies compare small numbers of methods (Tijsma et al., 2016; White & White, 2010). Kumaraswamy et al. (2018) compare a larger set of algorithms, different from the ones we study in this work, but to showcase their newly proposed algorithm rather than to understand the underlying strategies of the compared methods.

The goal of this work is to begin a systematic conceptual categorization of exploration methods that is supported by empirical results in different types of domains. We select domains to highlight particular properties that make exploration difficult, for example difficult transition or reward structures. While this categorization will necessarily be incomplete for both methods and domains, our goal is to provide a foundation for future studies on exploration.

We first propose a categorization of exploration techniques based on their underlying concepts. We then describe a novel suite of environments that have been designed to isolate specific aspects of exploration, including the effects of misleading rewards, large state spaces, and dynamics that make it difficult to reach large rewards. We also provide experiments comparing representative exploration methods from each category. Each agent uses an exploration method in conjunction with Q-learning (Watkins, 1989) with linear function approximation. The agents learn for a fixed number of steps and are evaluated offline to determine the quality of their learned policies. Though we restrict attention to linear function approximation in this first study, we compare methods designed for neural networks such as DQN (Mnih et al., 2015), and focus investigation on methods that scale to larger problems.

Our empirical results largely support our proposed categorization: methods in the same category tend to fail or succeed similarly depending on the properties of the environment. We also find that no single exploration method performs better than the others across all environments, suggesting that novel algorithms should be tested on suites of environments that provide interpretable, multifaceted views into the algorithms' behaviour.

[1] Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada [2] Google DeepMind, Edmonton, Alberta, Canada. Correspondence to: Niko Yasui <yasui@ualberta.ca>.

*Proceedings of the $2^{nd}$ Exploration in Reinforcement Learning Workshop at the $36^{th}$ International Conference on Machine Learning*, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

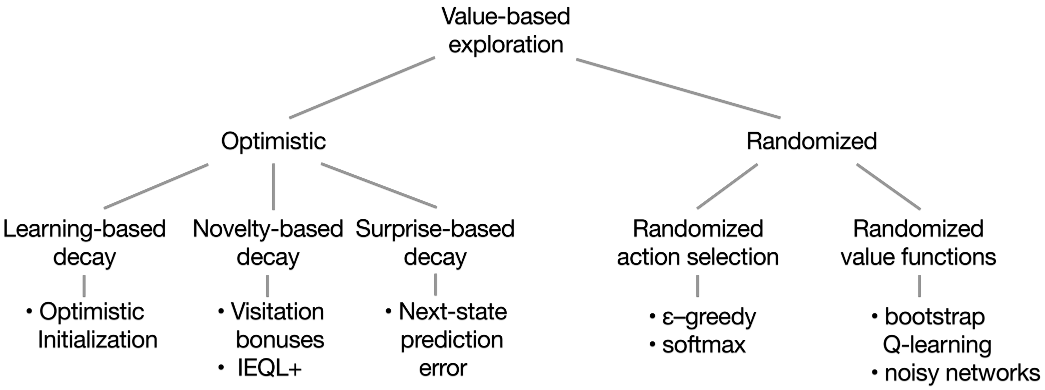

*Figure 1.* Categorization of value-based reinforcement learning control methods. Bullet points indicate methods that fall into each category.

### 1.1. Overview of exploration methods

Value-based exploration methods fall into two main categories that are distinguished by the way they treat uncertainty. Methods in the first category are Optimistic in the Face of Uncertainty (OFU): these methods behave according to a plausible estimate of the most rewarding MDP given the observed data. Methods in the second category are Randomized; they sample from distributions over optimal actions, value functions, or MDPs that are plausible given the observed data.

When using linear function approximation, a simple but widely-used strategy is to initialize each of the weights to some optimistic value. Then, on average, the estimated value of the visited state-action pair decreases at each time-step, and the agent chooses another action when it visits similar states. Optimistic initialization is an example of an OFU method whose uncertainty decays based on the agent's value-learning process.

Uncertainty can also decay based on novelty, or the number of visits to a state or state-action pair. In the tabular setting, Meuleau & Bourgine (1999) proposed a model-free method called IEQL+, based on the Interval Estimation algorithm for bandits (Kaelbling, 1993). IEQL+ backs up uncertainty estimates through the value function by adding a confidence-interval inspired novelty bonus to the reward. Bellemare et al. (2016) describe a similar algorithm for non-linear function approximation, which combines DQN with these count-based reward bonuses. Since identifying states is generally impossible when using function approximation, they created a visitation density model over the raw state space to approximate state visitations. Several other methods also use count-based reward bonuses and DQN; Martin et al. (2017) propose a density model over linear features rather than the raw state, while others from Tang et al. (2017) or Abel et al. (2016) respectively hash or cluster states before counting them. Fox, Choshen, and Loewenstein (2018) count state visitation using a function similar to the value function. These methods are all based on giving the agent a reward bonus proportional to some measure of novelty.

While count-based exploration methods use a reward bonus based on novelty, Stadie, Levine, and Abbeel (2015) and Pathak et al. (2017) propose reward bonuses based on surprise, using next-state prediction error. These methods learn a model to predict the next state, and encourage exploration for states with high prediction error. In using prediction error as a reward bonus, these models implicitly assume that the accuracy of the value function and state-prediction function are closely related.

The second category contains randomized methods, which randomize either action selection and or the value function. One of the most popular action-selection methods, $\epsilon$-greedy action selection (Watkins, 1989), selects the action with the highest estimated value, called the greedy action, with probability $1-\epsilon$, and a random action with probability $\epsilon$. Notably, DQN — which is the base algorithm for most recent count-based reward bonus methods — includes $\epsilon$-greedy action selection. Like $\epsilon$-greedy, softmax (Luce, 1959) also selects actions according to a distribution. The probability of selecting an action is the output of a softmax over the action-value estimates. While both these action-selection techniques are simple, they are among the oldest and most popular ways to explore.

Instead of randomizing action selection, some methods randomize their value functions and act greedily according to a sampled value function. One such method, Bayesian dropout (Gal & Ghahramani, 2016), represents a distribution over value functions using a single neural network by randomly excluding nodes from the value calculation. Another method, Bootstrap DQN (Osband et al., 2016), represents a distribution using many separate value functions. Each value function is updated to a different extent every time-step according to a sampled bootstrap parameter. Bootstrap DQN explores consistently during an episode by acting according to the same value function for the entire episode. Finally, Fortunato et al. (2018) propose noisy networks, which learn a random value function parameterized by mean and scaled standard normal noise terms.

### 1.2. Experimental setup

Our experiments are designed around components of an environment that affect an agent's performance. Since a reinforcement learning environment is primarily defined by the reward and transition dynamics, each of our environments makes one or both of the components difficult. Complete descriptions of the environments are available in the Appendices. The difficulty can come from stochastic rewards or transitions, as in VarianceWorld or continuous RiverSwim respectively, or from locally misleading reward signals, as in Antishaping. The environment can also be difficult simply because the state-action space is large, as in the five-dimensional Hypercube. The latter two environments are inspired by the RL Acid (Langford, 2018) set of difficult problems for reinforcement learning. Continuous RiverSwim is an adaptation of Strehl & Littman's 2008 Riverswim. We advocate the use of our environments as benchmarks; they pose significant challenges for current control methods, while remaining interpretable in their simplicity.

We implement agents from each of the conceptual categories using Q-learning. Optimistic initialization represents the class of OFU methods with learning-based uncertainty decay. Our novelty-based uncertainty agents use reward bonuses that decay with the square root of state-counts or state-action counts. We also include a linear version of IEQL+, which combines optimistic initialization and state-action count based reward bonuses. Including IEQL+ allows us to observe the joint effect of optimistic initialization and count-based reward bonuses.

We also include a method with a reward bonus that is based on the error of a linear next-state prediction function. The function independently predicts each dimension of the raw state using the same tile-coded features and linear architecture as the value function. This setup allows the next-state learning process to mirror the value function's learning process.

To represent the randomized methods, we first include $\epsilon$-greedy and softmax action selection as simple extensions of Q-learning. We use bootstrap Q-learning and a linear version of a noisy network to represent methods with randomized value functions. Our bootstrap Q-learning learns 10 value functions in parallel, whose updates are weighted by independent samples from a $\mathrm{Poi}(1)$ distribution. Our noisy network is identical to Q-learning, except that the agent learns parameters for both the value and noise terms. Finally, we include two agents that are not strictly part of the comparison, but that will provide intuition about the behaviour of other agents: a random agent for a baseline, and an actor-critic agent with a softmax policy, to be roughly representative of policy gradient methods.

Each agent's features are represented by tile-coding (Sutton & Barto, 1998) with two tiles per input dimension and 32 tilings, according to the generally useful trick of using few tiles and many tilings. Agents are trained for 500k time-steps according to their exploration strategies, after which their target policies are evaluated without learning for 100k time-steps. We report the total reward accumulated by each agent during the evaluation phase. Further experimental details can be found in the Supplementary Material. Code can be found here.

## 2. Results

In general, our results support the conceptual categories we describe in Section 1.1. The randomized value function methods performed almost identically in each of the environments, even though their value function representations and sampling frequencies are very different. Both methods were able to learn a good policy in the Antishaping experiment, but performed at least as badly as the random agent in the other environments. In Hypercube, the distributional value function agents found policies that only touch one wall, which is worse than randomly picking actions. In Osband et al. (2018), BootstrapDQN performed well on an environment that is similar to RiverSwim called DeepSea. We suspect that the use of experience replay (see Van Seijen & Sutton, 2015) may contribute to the improved performance observed in that work.

By contrast, the action-sampling methods behaved quite diversely. In Antishaping and RiverSwim, the small magnitude of the misleading reward signals meant that the softmax action probabilities remained close to uniform random until the larger rewards were found, effectively countering the misleading reward. The policy followed by $\epsilon$-greedy was greatly affected by the small misleading reward signals. However, in Hypercube the $\epsilon$-greedy agent's greedy action tended to push the agent into an edge that it was already touching, concentrating exploration in valuable states.

The OFU methods are not as cleanly delineated into conceptual categories as the sampling methods. Surprise and novelty-based uncertainty decay methods performed quite similarly, with results that are consistent with promoting exploration of the entire state or state-action space. In general, surprise-based uncertainty methods do not promote sweeping the entire state-space. However, the next-state prediction error in our experiments may have decayed at a similar rate to the state space visitation, producing similar

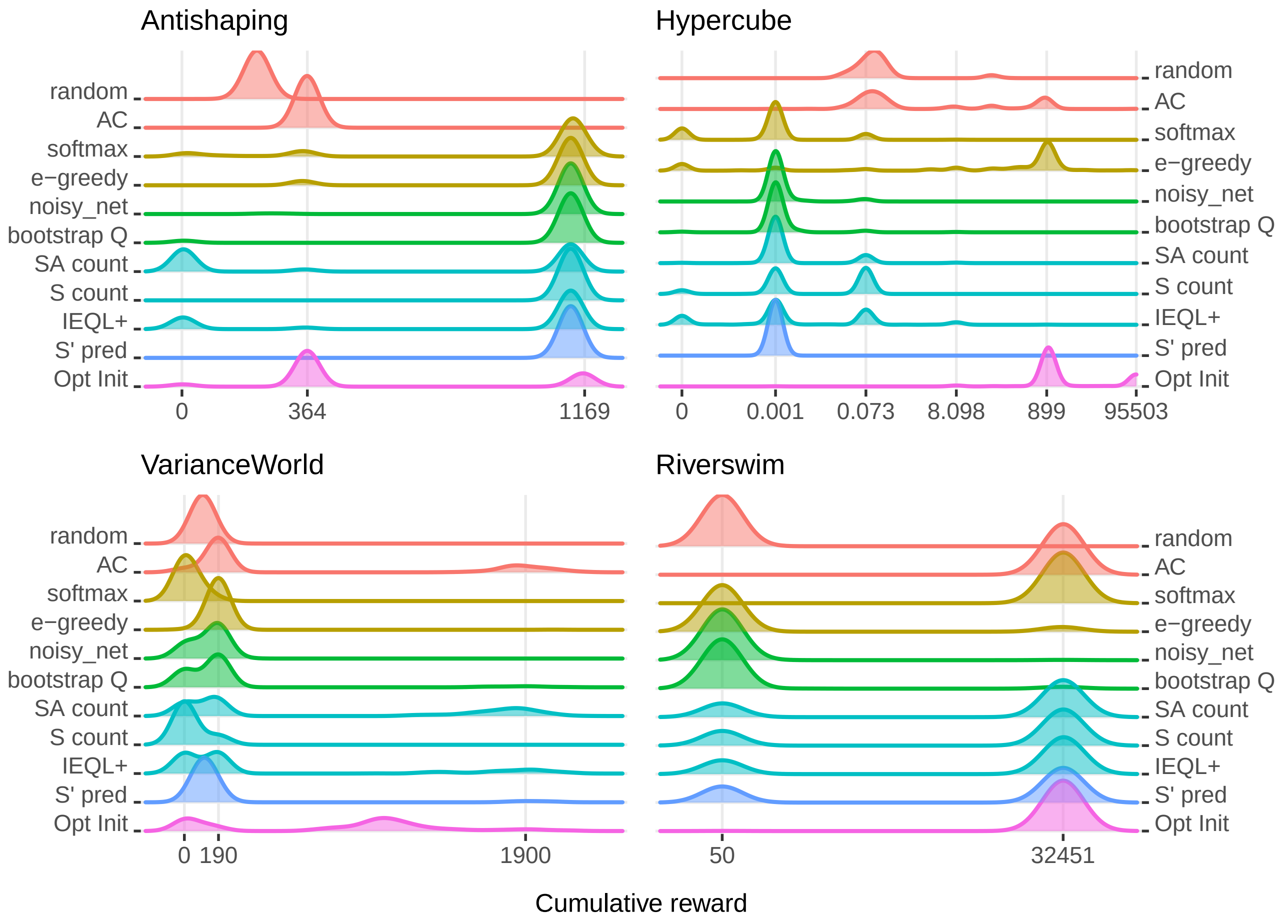

*Figure 2.* The distributions of total reward collected by each agent over 360 runs of the 100k time-step evaluation phase. Agents are colored based on their category, with the first two methods, random actions and Actor Critic, being baselines. More density over larger reward values (farther right) indicates improved performance. Vertical lines correspond to the reward obtained by different policies: in Antishaping, the policies of staying in the center or of moving to the nearest environment boundary; in Hypercube, policies that touch increasing numbers of edges; in VarianceWorld and RiverSwim, the policies that go to the smaller or larger reward.

patterns of behaviour.

Policies found by optimistic initialization behaved quite differently from those found by the other OFU methods. Even IEQL+, which uses optimistic initialization in conjunction with reward bonuses based on state-action counts, found policies that were indistinguishable from the other novelty-based methods. Optimistic initialization did not perform well in Antishaping and VarianceWorld because the optimistic values decayed too quickly for the agent to learn from the challenging reward functions. In the other two environments, optimistic initialization was able to find very strong policies.

## 3. Take-home messages

**A broadly successful exploration heuristic has not yet been identified.** Every agent we tested performed at or below the level of random action selection on at least two of the four domains.

**Optimistic initialization and $\epsilon$-greedy are the most promising options for practical problems with large state-action spaces.** In Hypercube, the rest of the agents performed similarly to or worse than random action selection.

**The ideal exploration heuristic depends on properties of the environment.** Applied researchers can tailor practical exploration strategies to the properties of their target environments.

## References

Abel, D., Agarwal, A., Diaz, F., Krishnamurthy, A., and Schapire, R. E. Exploratory gradient boosting for reinforcement learning in complex domains. *arXiv preprint arXiv:1603.04119*, 2016.

Bellemare, M. G., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pp. 1471–1479, 2016.

Fortunato, M., Azar, M. G., Piot, B., Menick, J., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., Blundell, C., and Legg, S. Noisy Networks for Exploration. *Proceedings of the Sixth International Conference on Learning Representations*, 2018.

Fox, L., Choshen, L., and Loewenstein, Y. DORA the explorer: Directed outreaching reinforcement action-selection. In *International Conference on Learning Representations*, 2018.

Gal, Y. and Ghahramani, Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059, 2016.

Kaelbling, L. P. *Learning in embedded systems*. MIT press, 1993.

Kumaraswamy, R., Schlegel, M., White, A., and White, M. Context-dependent upper-confidence bounds for directed exploration. In *Advances in Neural Information Processing Systems*, 2018.

Langford, J. RL Acid. Retrieved from https://github.com/JohnLangford/RL_acid, 2018.

Luce, R. D. *Individual Choice Behavior*. Wiley, New York, 1959.

Martin, J., Sasikumar, S. N., Everitt, T., and Hutter, M. Count-based exploration in feature space for reinforcement learning. *arXiv preprint arXiv:1706.08090*, 2017.

Meuleau, N. and Bourgine, P. Exploration of multi-state environments: Local measures and back-propagation of uncertainty. *Machine Learning*, 35(2):117–154, 1999.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529, 2015.

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. Deep exploration via bootstrapped dqn. In *Advances in neural information processing systems*, pp. 4026–4034, 2016.

Osband, I., Aslanides, J., and Cassirer, A. Randomized prior functions for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 8617–8629, 2018.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML)*, volume 2017, 2017.

Stadie, B. C., Levine, S., and Abbeel, P. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015.

Strehl, A. L. and Littman, M. L. An analysis of model-based interval estimation for Markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Szita, I. and Lőrincz, A. The many faces of optimism: a unifying approach. In *Proceedings of the 25th international conference on Machine learning*, pp. 1048–1055. ACM, 2008.

Tang, H., Houthooft, R., Foote, D., Stooke, A., Chen, X., Duan, Y., Schulman, J., DeTurck, F., and Abbeel, P. # exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 2753–2762, 2017.

Thrun, S. B. Efficient Exploration In Reinforcement Learning. Technical report, 1992.

Tijsma, A. D., Drugan, M. M., and Wiering, M. A. Comparing exploration strategies for q-learning in random stochastic mazes. In *IEEE Symposium Series on Computational Intelligence*, pp. 1–8. IEEE, 2016.

Van Seijen, H. and Sutton, R. A deeper look at planning as learning from replay. In *International conference on machine learning*, pp. 2314–2322, 2015.

Watkins, C. J. C. H. *Learning from delayed rewards*. PhD thesis, King's College, Cambridge, 1989.

White, M. and White, A. Interval estimation for reinforcement-learning algorithms in continuous-state domains. In *Advances in Neural Information Processing Systems*, 2010.