BOOSTING REINFORCEMENT LEARNING WITH EXTREMUM EXPERIENCES

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning research has achieved high acceleration in its progress starting from the initial installation of deep neural networks as function approximators to learn policies that make sequential decisions in high-dimensional state representation MDPs. While several consecutive barriers have been broken in deep reinforcement learning research (i.e. learning from high-dimensional states, learning purely via self-play), several others still stand. On this line, in our paper we focus on experience collection in high-dimensional complex MDPs and we propose a unique technique based on experiences obtained through extremum actions. Our method provides theoretical basis for efficient experience collection, and further comes with zero additional computational cost while leading to significant sample efficiency gains in deep reinforcement learning training. We conduct extensive experiments in the Arcade Learning Environment with high-dimensional state representation MDPs. We demonstrate that our technique improves the human normalized median scores of Arcade Learning Environment by 248% in the low-data regime.

1 INTRODUCTION

Utilization of deep neural networks as function approximators enabled learning functioning policies in high-dimensional state representation MDPs (Mnih et al., 2015). Following this initial work, the current line of work trains deep reinforcement learning policies to solve highly complex problems from game solving (Hasselt et al., 2016; Schrittwieser et al., 2020) to self driving vehicles (Lan et al., 2020). Yet there are still remaining unsolved problems restricting the current capabilities of deep neural policies. One of the main intrinsic open problems in deep reinforcement learning research is experience collection and sample complexity in high-dimensional state representation MDPs. While prior work extensively studied the exploration problem in bandits and tabular reinforcement learning, and proposed various algorithms and techniques optimal to the tabular form or the bandit setting (Kearns & Singh, 2002; Brafman & Tennenholtz, 2002; Karnin et al., 2013; Lu & Roy, 2019), experience collection in deep reinforcement learning remains an open challenging problem while practitioners repeatedly employ quite simple yet effective techniques (i.e. ϵ-greedy) (Flennerhag et al., 2022; Hasselt et al., 2016; Wang et al., 2016; Hamrick et al., 2020).

Despite the provable optimality of these techniques in the tabular or bandit setting, they generally rely strongly on the assumptions of tabular reinforcement learning, and in particular on the ability to record tables of statistical estimates for every state-action pair which have size growing with the number of states times the number of actions. Hence, these assumptions are far from what is being faced in the deep reinforcement learning setting where states and actions can be parametrized by high-dimensional representations. Thus, in high-dimensional complex MDPs, for which deep neural networks are used as function approximators, the efficiency and the optimality of the methods proposed for tabular settings do not transfer well to deep reinforcement learning experience collection. Hence, in deep reinforcement learning research still, naive and standard techniques (e.g. ϵ-greedy) are preferred over both the optimal tabular techniques and over the particular recent experience collection techniques targeting only high scores for particular games (Mnih et al., 2015; Hasselt et al., 2016; Wang et al., 2016; Anschel et al., 2017; Bellemare et al., 2017; Dabney et al., 2018; Lan et al., 2020; Flennerhag et al., 2022).

Sample efficiency in deep neural policies is still one of the main challenging problems restricting research progress in reinforcement learning. The magnitude of the number of samples required to learn and adapt continuously is one of the main limiting factors preventing current state-of-the-art deep reinforcement learning algorithms from being deployed in many diverse settings, but most importantly one of the main challenges that needs to be dealt with on the way to building neural policies that can generalize and adapt continuously in non-stationary environments. In our paper we aim to seek answers for the following questions:

- *Can we collect experiences in a high-dimensional state representation MDP more efficiently with zero additional computational cost?*
- *Is there a natural theoretical motivation that can be used to design a zero-cost exploration strategy while achieving high sample efficiency?*

To be able to answer these questions, in our paper we focus on environment interactions in deep reinforcement learning and make the following contributions:

- We propose a novel experience collection technique based on minimizing the state-action value function to increase the information gain from each particular experience acquired in the MDP.
- We conduct an extensive study in the Arcade Learning Environment 100K benchmark with the state-of-the-art algorithms and demonstrate that our temporal difference learning algorithm improves performance by 248% across the entire benchmark compared to the baseline algorithm.
- We demonstrate the efficacy of our proposed MaxMin TD Learning algorithm in terms of sample-efficiency. Our method based on maximizing novel experiences via minimizing the state-action value function reaches approximately to the same performance level as model-based deep reinforcement learning algorithms, without building and learning any model of the environment.

2 Background and Preliminaries

The reinforcement learning problem is formalized as a Markov Decision Process (MDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, r, \gamma, \rho_0, \mathcal{P} \rangle$ that contains a continous set of states $s \in \mathcal{S}$, a set of discrete actions $a \in \mathcal{A}$, a probability transition function $\mathcal{T}(s, a, s')$ on $\mathcal{S} \times \mathcal{A} \times \mathcal{S}$, discount factor γ, a reward function $r(s,a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ with initial state distribution ρ_0. A policy $\pi(s,a) : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ in an MDP is a mapping function between states and actions assigning a probability distribution over actions for each state $s \in \mathcal{S}$. The main goal in reinforcement learning is to learn an optimal policy π that maximizes the discounted expected cumulative rewards.

$$\mathcal{R} = \mathbb{E}_{a_t \sim \pi(s_t, \cdot)} \sum_t \gamma^t r(s_t, a_t),$$

where $a_t \sim \pi(s_t, \cdot)$. In Q-learning the learned policy is parameterized by a state-action value function $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, which represents the value of taking action a in state s. The optimal state-action value function is learnt via iterative Bellman update

$$Q(s_t, a_t) = r(s_t, a_t) + \gamma \sum_{s_t} \mathcal{T}(s_t, a_t, s_{t+1}) \mathcal{V}(s_{t+1}).$$

where $\mathcal{V}(s_{t+1}) = \max_a Q(s_{t+1}, a)$. Let a^* be the action maximizing the state-action value function, $a^*(s) = \arg\max_a Q(s,a)$, in state s. Once the Q-function is learnt the policy is determined via taking action $a^*(s) = \arg\max_a Q(s,a)$. In deep reinforcement learning, the state space or the action space is large enough that it is not possible to learn and store the state-action values in a tabular form. Thus, the Q-function is approximated via deep neural networks.

$$\theta_{t+1} = \theta_t + \alpha(r(s_t, a_t) + \gamma Q(s_{t+1}, \arg\max_a Q(s_{t+1}, a; \theta_t); \theta_t) - Q(s_t, a_t; \theta_t)) \nabla_{\theta_t} Q(s_t, a_t; \theta_t)$$

In deep double-Q learning, two Q-networks are used to decouple the Q-network deciding which action to take and the Q-network to evaluate the action taken.

$$\theta_{t+1} = \theta_t + \alpha(r(s_t, a_t) + \gamma Q(s_{t+1}, \arg\max_a Q(s_{t+1}, a; \theta_t); \hat{\theta}_t) - Q(s_t, a_t; \theta_t)) \nabla_{\theta_t} Q(s_t, a_t; \theta_t)$$

Current deep reinforcement learning algorithms use ϵ-greedy exploration during training (Wang et al., 2016; Mnih et al., 2015; Hasselt et al., 2016; Hamrick et al., 2020; Flennerhag et al., 2022). In particular, the ϵ-greedy algorithm takes an action $a_k \sim \mathcal{U}(\mathcal{A})$ with probability ϵ in a given state s, i.e. $\pi(s, a_k) = \frac{\epsilon}{|\mathcal{A}|}$, and takes an action $a^* = \arg\max_a Q(s,a)$ with probability $1-\epsilon$, i.e.

$$\pi(s, \arg\max_a Q(s,a)) = 1 - \epsilon + \frac{\epsilon}{|\mathcal{A}|}$$

While a family of algorithms have been proposed based on counting state visitations (i.e. the number of times action a has been taken in state s by time step t) with provable optimal regret bounds using the principal of optimism in the face of uncertainty in the tabular MDP setting, yet incorporating these count-based methods in high-dimensional state representation MDPs requires substantial complexity including training additional deep neural networks to estimate counts or other uncertainty metrics. As a result, many state-of-the-art deep reinforcement learning algorithms still use simple, randomized experience collection methods based on sampling a uniformly random action with probability ϵ (Mnih et al., 2015; Hasselt et al., 2016; Wang et al., 2016; Hamrick et al., 2020; Flennerhag et al., 2022), or the injection of random noise via noisy-networks (Hessel et al., 2018). Nonetheless, we still provide comparison to count-based methods in Section 4 and Section 6.

3 BOOSTING TEMPORAL DIFFERENCE

In deep reinforcement learning the state-action value function is initialized with random weights (Mnih et al., 2015; 2016; Hasselt et al., 2016; Wang et al., 2016; Schaul et al., 2016; Oh et al., 2020; Schrittwieser et al., 2020; Hubert et al., 2021). Thus, in the early phase of the training the Q-function will behave more like a random function rather than providing an accurate representation of the optimal state-action values. In particular, early in training the Q-function, on average, will assign approximately similar values to states that are similar, and will have little correlation with the immediate rewards. We first formalize this intuition in the following definitions.

Definition 3.1 (η-uninformed Q). Let $\eta > 0$. A Q-function parameterized by weights $\theta \sim \Theta$ is η-uninformed if for any state $s \in \mathcal{S}$ with $a^{\text{min}} = \arg\min_a Q_\theta(s,a)$ we have

$$|\mathbb{E}_{\theta\sim\Theta}[r(s_t, a^{\text{min}})] - \mathbb{E}_{a\sim\mathcal{U}(\mathcal{A})}[r(s_t, a)]| < \eta.$$

Definition 3.2 (δ-smooth Q). Let $\delta > 0$. A Q-function parameterized by weights $\theta \sim \Theta$ is δ-smooth if for any state $s \in \mathcal{S}$ and action $\hat{a} = \hat{a}(s,\theta)$ with $s' \sim \mathcal{T}(s, \hat{a}, \cdot)$ we have

$$|\mathbb{E}_{s'\sim\mathcal{T}(s,\hat{a},\cdot),\theta\sim\Theta}[\max_a Q_\theta(s,a)] - \mathbb{E}_{s'\sim\mathcal{T}(s,\hat{a},\cdot),\theta\sim\Theta}[\max_a Q_\theta(s',a)]| < \delta$$

where the expectation is over both the random initialization of the Q-function weights, and the random transition to state $s' \sim \mathcal{T}(s, \hat{a}, \cdot)$.

Definition 3.3 (Disadvantage Gap). For a state-action value function Q_θ the disadvantage gap in a state $s \in \mathcal{S}$ is given by

$$\mathcal{D}(s) = \mathbb{E}_{a\sim\mathcal{U}(\mathcal{A}),\theta\sim\Theta}[Q_\theta(s,a) - Q_\theta(s, a^{\text{min}})]$$

where $a^{\text{min}} = \arg\min_a Q_\theta(s,a)$.

The following proposition captures the intuition that when the Q-function on average assigns similar maximum values to consecutive states, choosing the action minimizing the state-action value function will achieve an above-average temporal difference.

Proposition 3.4. *Let $\eta, \delta > 0$ and suppose that $Q_\theta(s,a)$ is η-uninformed and δ-smooth. Let $s_t \in \mathcal{S}$ be a state, and let a^{min} be the action minimizing the state-action value in a given state s_t, $a^{min} = \arg\min_a Q_\theta(s_t, a)$. Let $s^{min}_{t+1} \sim \mathcal{T}(s_t, a^{min}, \cdot)$. Then for an action $a_t \sim \mathcal{U}(\mathcal{A})$ with $s_{t+1} \sim \mathcal{T}(s_t, a_t, \cdot)$ we have*

$$\begin{aligned}
&\mathbb{E}_{s^{min}_{t+1}\sim\mathcal{T}(s_t,a^{min},\cdot),\theta\sim\Theta}[r(s_t, a^{min}) + \gamma \max_a Q_\theta(s^{min}_{t+1}, a) - Q_\theta(s_t, a^{min})] \\
&\qquad > \mathbb{E}_{a_t\sim\mathcal{U},(\mathcal{A})s_{t+1}\sim\mathcal{T}(s_t,a_t,\cdot),\theta\sim\Theta}[r(s_t, a_t) + \gamma \max_a Q_\theta(s_{t+1}, a) - Q_\theta(s_t, a_t)] \\
&\qquad + \mathcal{D}(s) - 2\delta - \eta
\end{aligned}$$

Proof. Since $Q_\theta(s,a)$ is δ-smooth we have

$$\begin{aligned}
&\mathbb{E}_{s^{\text{min}}_{t+1}\sim\mathcal{T}(s_t,a^{\text{min}},\cdot),\theta\sim\Theta}[\gamma \max_a Q_\theta(s^{\text{min}}_{t+1},a) - Q_\theta(s_t,a_{\text{min}})] \\
&\qquad > \gamma\mathbb{E}_{\theta\sim\Theta}[\max_a Q_\theta(s_t,a)] - \delta - \mathbb{E}_{\theta\sim\Theta}[Q_\theta(s_t,a_{\text{min}})] \\
&\qquad > \gamma\mathbb{E}_{s_{t+1}\sim\mathcal{T}(s_t,a_t,\cdot),\theta\sim\Theta}[\max_a Q_\theta(s_{t+1},a)] - 2\delta - \mathbb{E}_{\theta\sim\Theta}[Q_\theta(s_t,a_{\text{min}})] \\
&\qquad \geq \mathbb{E}_{a_t\sim\mathcal{U}(\mathcal{A}),s_{t+1}\sim\mathcal{T}(s_t,a_t,\cdot),\theta\sim\Theta}[\gamma \max_a Q_\theta(s_{t+1},a) - Q_\theta(s_t,a_t)] \\
&\qquad + \mathcal{D}(s) - 2\delta
\end{aligned}$$

where the last line follows from Definition 3.3. Further, because $Q_\theta(s,a)$ is η-uninformed,

$$\mathbb{E}_{\theta\sim\Theta}[r(s_t,a^{\text{min}})] > \mathbb{E}_{a_t\sim\mathcal{U}(\mathcal{A})}[r(s_t,a_t)] - \eta.$$

Combining with the previous inequality completes the proof. □

In words, the proposition shows that the temporal difference achieved by the minimum-value action is above-average by an amount approximately equal to the disadvantage gap. The above argument can be extended to the case where action selection and evaluation in the temporal difference are computed with two different sets of weights θ and $\hat{\theta}$ as in double Q-learning.

Definition 3.5 (δ-smoothness for Double-Q)**.** Let $\delta > 0$. A pair of Q-functions parameterized by weights $\theta \sim \Theta$ and $\hat{\theta} \sim \Theta$ are δ-smooth if for any state $s \in \mathcal{S}$ and action $\hat{a} = \hat{a}(s,\theta) \in \mathcal{A}$ with $s' \sim \mathcal{T}(s,\hat{a},\cdot)$ we have

$$\begin{aligned}
\Bigg|&\mathbb{E}_{s'\sim\mathcal{T}(s,\hat{a},\cdot),\theta\sim\Theta,\hat{\theta}\sim\Theta}\left[Q_{\hat{\theta}}(s,\arg\max_a Q_\theta(s,a))\right] \\
&\qquad\qquad - \mathbb{E}_{s'\sim\mathcal{T}(s,\hat{a},\cdot),\theta\sim\Theta,\hat{\theta}\sim\Theta}\left[Q_{\hat{\theta}}(s',\arg\max_a Q_\theta(s',a))\right]\Bigg| < \delta
\end{aligned}$$

where the expectation is over both the random initialization of the Q-function weights θ and $\hat{\theta}$, and the random transition to state $s' \sim \mathcal{T}(s,\hat{a},\cdot)$.

With this definition we can then prove that choosing the minimum valued action will lead to a temporal difference that is above-average by approximately $\mathcal{D}(s)$.

Proposition 3.6. *Let $\eta, \delta > 0$ and suppose that Q_θ and $Q_{\hat{\theta}}$ are η-uniformed and δ-smooth. Let $s_t \in \mathcal{S}$ be a state, and let $a^{min} = \arg\min_a Q_\theta(s_t,a)$. Let $s^{min}_{t+1} \sim \mathcal{T}(s_t,a^{min},\cdot)$. Then for an action $a_t \sim \mathcal{U}(\mathcal{A})$ with $s_{t+1} \sim \mathcal{T}(s_t,a_t,\cdot)$ we have*

$$\begin{aligned}
&\mathbb{E}_{s_{t+1}\sim\mathcal{T}(s,a,\cdot),\theta\sim\Theta,\hat{\theta}\sim\Theta}[r(s_t,a^{min}) + \gamma Q_{\hat{\theta}}(s^{min}_{t+1},\arg\max_a Q_\theta(s^{min}_{t+1},a)) - Q_\theta(s_t,a^{min})] \\
&> \mathbb{E}_{a_t\sim\mathcal{U}(\mathcal{A}),s_{t+1}\sim\mathcal{T}(s,a,\cdot),\theta\sim\Theta,\hat{\theta}\sim\Theta}[r(s_t,a_t) + \gamma Q_{\hat{\theta}}(s_{t+1},\arg\max_a Q_\theta(s_{t+1},a)) - Q_\theta(s_t,a_t)] \\
&\qquad + \mathcal{D}(s) - 2\delta - \eta
\end{aligned}$$

Proof. Since Q_θ and $Q_{\hat{\theta}}$ are δ-smooth we have

$$\begin{aligned}
&\mathbb{E}_{s^{\text{min}}_{t+1}\sim\mathcal{T}(s_t,a^{\text{min}},\cdot),\theta\sim\Theta,\hat{\theta}\sim\Theta}[\gamma Q_{\hat{\theta}}(s^{\text{min}}_{t+1},\arg\max_a Q_\theta(s^{\text{min}}_{t+1},a)) - Q_\theta(s_t,a^{\text{min}})] \\
&\qquad > \mathbb{E}_{s^{\text{min}}_{t+1}\sim\mathcal{T}(s_t,a^{\text{min}},\cdot),\theta\sim\Theta,\hat{\theta}\sim\Theta}[\gamma Q_{\hat{\theta}}(s_t,\arg\max_a Q_\theta(s_t,a)) - Q_\theta(s_t,a^{\text{min}})] - \delta \\
&\qquad > \mathbb{E}_{s_{t+1}\sim\mathcal{T}(s_t,a_t,\cdot),\theta\sim\Theta,\hat{\theta}\sim\Theta}[\gamma Q_{\hat{\theta}}(s_{t+1},\arg\max_a Q_\theta(s_{t+1},a)) - Q_\theta(s_t,a^{\text{min}})] - 2\delta \\
&\qquad \geq \mathbb{E}_{s_{t+1}\sim\mathcal{T}(s_t,a_t,\cdot),\theta\sim\Theta,\hat{\theta}\sim\Theta}[\gamma Q_{\hat{\theta}}(s_{t+1},\arg\max_a Q_\theta(s_{t+1},a)) - Q_\theta(s_t,a_t)] \\
&\qquad + \mathcal{D}(s) - 2\delta
\end{aligned}$$

where the last line follows from Definition 3.3. Further, because Q_θ and $Q_{\hat{\theta}}$ are η-uniformed,

$$\mathbb{E}_{\theta\sim\Theta,\hat{\theta}\sim\Theta}[r(s_t,a^{\text{min}})] > \mathbb{E}_{a_t\sim\mathcal{U}(\mathcal{A})}[r(s_t,a_t)] - \eta.$$

Combining with the previous inequality completes the proof. □

At first, the results in Proposition 3.4 and 3.6 might appear counterintuitive. The fact that the Q-function is δ-smooth and η-uninformed seem like properties of a random function. Thus, taking the minimum Q-value action should be approximately equivalent to taking a uniform random action. However, Proposition 3.4 and 3.6 show that the temporal difference achieved by taking the minimum action is larger than that of a random action by an amount equal to the disadvantage gap $\mathcal{D}(s)$. In order to reconcile these two statements it is useful at this point to look at the limiting case of the Q function at initialization. In particular, the following proposition shows that, at initialization, the distribution of the minimum value action in a given state is uniform by itself, but is constant once we condition on the weights θ.

Proposition 3.7. *Let θ be the random initial weights for the Q-function. For any state $s \in \mathcal{S}$ let $a^{min}(s) = \arg\min_{a' \in \mathcal{A}} Q_\theta(s, a')$. Then for any $a \in \mathcal{A}$*

$$\mathbb{P}_{\theta \sim \Theta}\left[\arg\min_{a' \in \mathcal{A}} Q_\theta(s, a') = a\right] = \frac{1}{|\mathcal{A}|}$$

i.e. the distribution $\mathbb{P}_{\theta \sim \Theta}[a^{min}(s)]$ is uniform. Simultaneously, the conditional distribution $\mathbb{P}_{\theta \sim \Theta}[a^{min}(s) \mid \theta]$ is constant.

Proof. Since $Q_\theta(s, \cdot)$ is a random function (given the random choice of θ), each action $a \in A$ is equally likely to be assigned the minimum Q-value in state s. Thus,

$$\mathbb{P}_{\theta \sim \Theta}\left[\arg\min_{a' \in \mathcal{A}} Q_\theta(s, a) = a\right] = \frac{1}{|\mathcal{A}|}.$$

However, given the value of θ, the value of $a^{\min}(s)$ is uniquely determined because

$$a^{\min}(s) = \arg\min_{a \in \mathcal{A}} Q_\theta(s, a).$$

Therefore, the distribution of $a^{\min}(s)$ conditional on θ is constant. □

This implies that, in states whose Q-values have not changed drastically from initialization, taking the minimum action is almost equivalent to taking a random action. However, while the action chosen early on in training is almost uniformly random when only considering the current state, it is at the same time completely determined by the current value of the weights θ. The temporal difference is also determined by the weights θ. Thus while the marginal distribution on actions taken is uniform, the temporal difference when taking the minimum action is quite different than from the case where an independently random action is chosen. In particular, in expectation over the random initialization $\theta \sim \Theta$, the temporal difference is higher when taking the minimum value action than that of a random action as demonstrated in Section 3. The main objective of our method is to increase the information gained from each experience via taking the actions that minimize the state-action value function. While minimization of the Q-function may initially be regarded as counterintuitive, Section 3 provides the exact theoretical justification on how taking actions that minimize the state-action value function results in higher temporal difference for the corresponding state transitions. Algorithm 1 summarizes our proposed algorithm MaxMin TD Learning based on minimizing the state-action value function as described in detail in Section 3. Note that populating the experience replay buffer and learning are happening simultaneously with different rates.

4 Motivating Example

As a motivating example we consider the chain MDP which consists of a chain of n states $s \in \mathcal{S} = \{1, 2, \cdots n\}$ each with four actions. Each state i has one action that transitions the agent up the chain by one step to state $i + 1$, one action that transitions the agent to state 2, one action that transitions the agent to state 3, and one action which resets the agent to state 1 at the beginning of the chain. All transitions have reward zero, except for the last transition returning the agent to the beginning from the n-th state. Thus, when started from the first state in the chain, the agent must learn a policy that takes $n - 1$ consecutive steps up the chain, and then one final step to reset and get the reward. For the chain MDP, we compare standard approaches to exploration in tabular Q-learning with our method MaxMin TD Learning based on minimization of the state-action values. In particular we compare

```
Algorithm 1: MaxMin TD Learning
Input: In MDP $\mathcal{M}$ with $\gamma \in (0,1]$, $s \in \mathcal{S}$, $a \in \mathcal{A}$ with $Q_\theta(s,a)$ function parametrized by $\theta$, $\mathcal{B}$
  experience replay buffer, $\epsilon$ exploration parameter, $\mathcal{N}$ is the training learning steps.
  Populating Experience Replay Buffer:
  for $s_t$ in $e$ do
      Sample $\kappa \sim U(0,1)$
      if $\kappa < \epsilon$ then
          $a^{min} = \arg\min_a Q(s_t, a)$
          $\mathcal{B} \leftarrow (r(s_t, a^{min}), s_t, s_{t+1}^{min}, a^{min})$
      else
          $a^* = \arg\max_a Q(s_t, a)$
          $\mathcal{B} \leftarrow (r(s_t, a^*), s_t, s_{t+1}, a^*)$
      end if
  end for
  Learning:
  for $n$ in $\mathcal{N}$ do
      Update with probability $\epsilon$:
      $\mathcal{TD} = r(s_t, a^{min})$
          $+\gamma \max_a Q(s_{t+1}^{min}, a) - Q(s_t, a^{min})$
      Update with probability $1-\epsilon$:
      $\mathcal{TD} = r(s_t, a^*)$
          $+\gamma \max_a Q(s_{t+1}, a) - Q(s_t, a^*)$
  end for
  $\nabla \mathcal{L}(\mathcal{TD})$
```

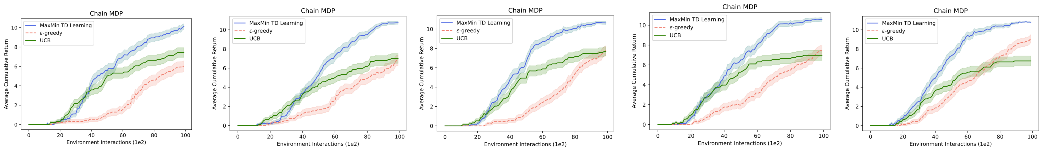


Figure 1: Learning curves in the chain MDP with our proposed algorithm MaxMin TD Learning, the canonical algorithm ϵ-greedy and the UCB algorithm with variations in ϵ.

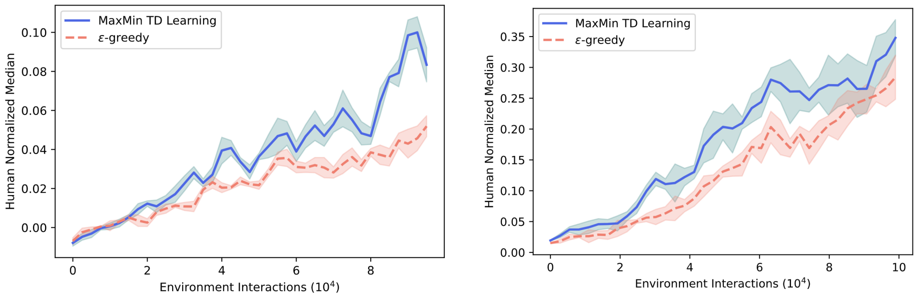
our method MaxMin TD Learning with both the ϵ-greedy action selection method, and the upper confidence bound (UCB) method. In more detail, in the UCB method the number of training steps t, and the number of times $N_t(s,a)$ that each action a has been taken in state s by step t are recorded. Furthermore, the action $a \in \mathcal{A}$ selection is determined as follows:

$$a^{\text{UCB}} = \underset{a \in \mathcal{A}}{\arg\max}\, Q(s,a) + 2\sqrt{\frac{\log t}{N_t(s,a)}}.$$

In a given state s if $N(s,a) = 0$ for any action a, then an action is sampled uniformly at random from the set of actions a' with $N(s,a') = 0$. For the experiments reported in our paper the length of the chain is set to $n = 10$. The Q-function is initialized by independently sampling each state-action value from a normal distribution with $\mu = 0$ and $\sigma = 0.1$. In each iteration we train the agent using Q-learning for 100 steps, and then evaluate the reward obtained by the argmax policy using the current Q-function for 100 steps. Note that the maximum achievable reward in 100 steps is 10. Figure 1 reports the learning curves for each method with varying $\epsilon \in [0.15, 0.25]$ with step size 0.025. The results in Figure 1 demonstrate that our method converges more quickly to the optimal policy than either of the standard approaches.

5 Large Scale Experimental Results

The experiments are conducted in the Arcade Learning Environment (ALE) (Bellemare et al., 2013). The Double-Q Network (Hasselt et al., 2016) initially proposed by (van Hasselt, 2010) is trained with prioritized experience replay (Schaul et al., 2016) without the dueling architecture with its original version (Hasselt et al., 2016). The experiments are conducted both in the 100K Arcade Learning Environment benchmark (van Hasselt et al., 2019), and the canonical version with 200 million frame training. Note that the 100K Arcade Learning Environment benchmark is an established baseline proposed to measure sample efficiency in deep reinforcement learning research. The ALE 100K benchmark contains 26 different Arcade Learning Environment games. The policies are evaluated after 100000 environment interactions. All of the polices in the experiments are trained over 5 random seeds. The hyperparameters and the architecture details are reported in the supplementary material. All of the results in the paper are reported with the standard error of the mean. The human normalized



Median

80th Percentile

Figure 2: Human normalized scores median and 80th percentile over all games in the Arcade Learning Environment (ALE) 100K benchmark for MaxMin TD Learning algorithm and the canonical exploration algorithm ϵ-greedy.

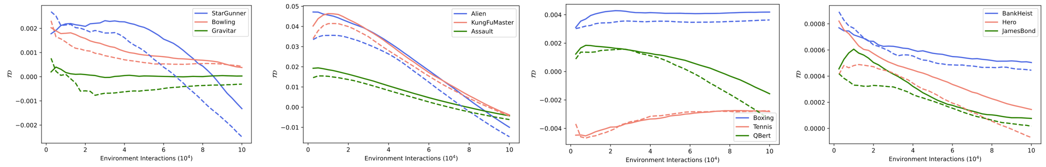


Figure 3: Temporal difference for our proposed algorithm MaxMin TD Learning and the canonical ϵ-greedy algorithm in the Arcade Learning Environment 100K benchmark. Dashed lines report the temporal difference for the ϵ-greedy algorithm and solid lines report the temporal difference for the MaxMin TD Learning algorithm. Colors indicate games.

Table 1: Human normalized scores median and 20th percentile across all of the games in the Arcade Learning Environment 100K benchmark for MaxMin TD Learning, ϵ-greedy and NoisyNetworks.

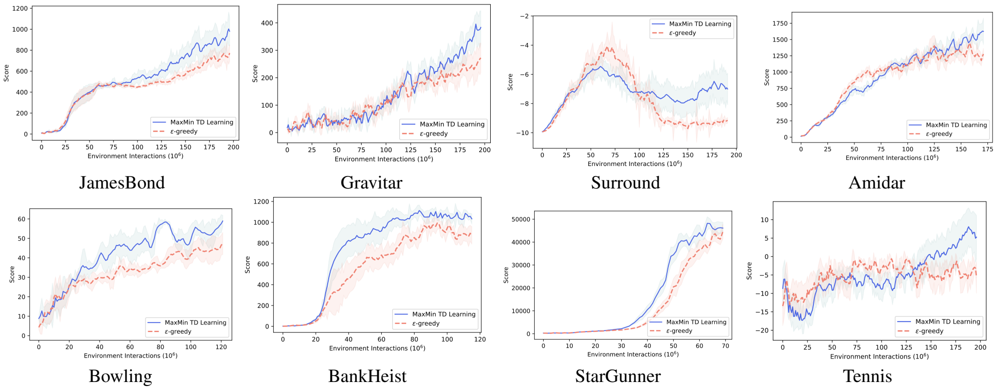
Method	MaxMin TD Learning	ϵ-greedy	NoisyNetworks
Human Normalized Median	**0.0927±0.0050**	0.0377±0.0031	0.0457±0.0035
20th Percentile	**0.0145±0.0003**	0.0056±0.0017	0.0102±0.0018
80th Percentile	**0.3762±0.0137**	0.2942±0.0233	0.1913±0.0144

scores are computed as,

$$\text{HN} = \frac{\text{Score}_{agent} - \text{Score}_{random}}{\text{Score}_{human} - \text{Score}_{random}} \tag{1}$$

For completeness we also report several results with 200 million frame training (i.e. 50 million environment interactions). In particular, Figure 4 demonstrates the learning curves for our proposed algorithm MaxMin TD Learning and the original version of the DDQN algorithm with ϵ-greedy training (Hasselt et al., 2016). In the large data regime we observe that while in some MDPs our proposed method MaxMin TD Learning that focuses on experience collection with novel temporal difference boosting via minimizing the state-action values converges faster, in other MDPs MaxMin TD Learning simply converges to a better policy. More concretely, while the learning curves of StarGunner, Bowling, JamesBond and BankHeist games in Figure 4 demonstrate the faster convergence rate of our proposed algorithm MaxMin TD Learning, the learning curves of the JamesBond, Amidar, BankHeist, Surround, Gravitar and Tennis games demonstrate that our experience collection technique not only increases the sample efficiency in deep reinforcement learning, but also results in learning a policy that is more close to optimal compared to learning a policy with the original method used in the DDQN algorithm.

Additionally, we also compare our proposed MaxMin TD Learning algorithm with NoisyNetworks as referred to in Section 2. Table 1 further demonstrates that the MaxMin TD Learning algorithm achieves significantly better performance results compared to NoisyNetworks. Furthermore, note that NoisyNetworks includes adding layers in the Q-network to increase exploration. However, this increases the number of parameters that have been added in the training process; thus, introducing additional cost to increase exploration. Table 1 reports results of human normalized median scores, 20th percentile, and 80th percentile for the Arcade Learning Environment 100K benchmark. Thus,


JamesBond Gravitar Surround Amidar

Bowling BankHeist StarGunner Tennis

Figure 4: The learning curves of StarGunner, Bowling, Surround, BankHeist, JamesBond, Amidar, Gravitar and Tennis with our proposed method MaxMin TD Learning and the ϵ-greedy algorithm in the Arcade Learning Environment with 200 million frame training.

Table 1 demonstrates that our proposed MaxMin TD Learning algorithm improves on the performance of the canonical algorithm ϵ-greedy by 248% and NoisyNetworks by 204%.

We further compare our proposed MaxMin TD Learning algorithm with another baseline algorithm QRDQN. In particular, Figure 5 reports results of human normalized median scores and 80th percentile over all of the games of the Arcade Learning Environment (ALE) in the low-data regime. These results once more demonstrate that the performance obtained by the MaxMin TD Learning algorithm is approximately double the performance achieved by the canonical experience collection techniques. As the results reported demonstrate, the MaxMin TD Learning algorithm achieves substantial sample-efficiency with zero-additional cost across many algorithms and different sample-complexity regions over canonical baseline alternatives.

6 Investigating the Temporal Difference

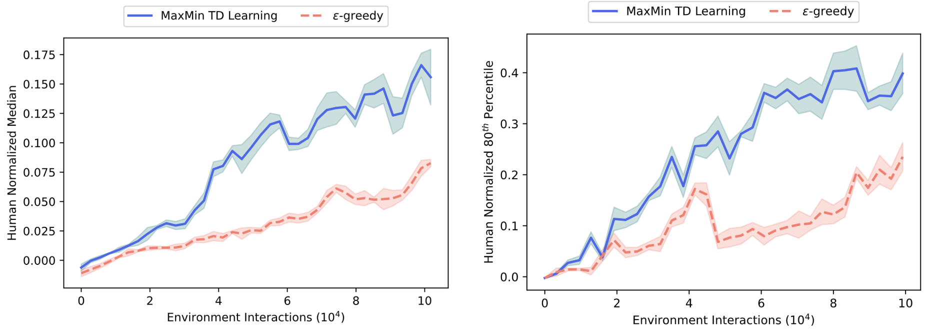
The original justification for exploring with the minimum Q-value action, is that taking this action tends to result in transitions with higher temporal difference. The theoretical analysis from Proposition 3.4 indicates that, when the Q function is δ-smooth and η-uninformed, taking the minimum value action results in an increase in the temporal difference proportional to the disadvantage gap. In particular, Proposition 3.4 states that the temporal difference achieved when taking the minimum Q-value action in state s exceeds the average temporal difference over a uniform random action by $\mathcal{D}(s) - 2\delta - \eta$. In order to evaluate how well the theoretical prediction matches reality, in this section we provide empirical measurements of the temporal difference in our experiments. To measure the change in the temporal difference when taking the minimum action versus the average action, we compare the temporal difference obtained by MaxMin TD Learning exploration with that obtained by ϵ-greedy exploration. In more detail, during training, for each batch Λ of transitions of the form (s_t, a_t, s_{t+1}) we record, the temporal difference

$$\begin{aligned}\mathcal{TD} &= \mathbb{E}_{(s_t,a_t,s_{t+1})\sim\Lambda}\mathcal{TD}(s_t, a_t, s_{t+1})\\ &= \mathbb{E}_{(s_t,a_t,s_{t+1})\sim\Lambda}[r(s_t, a_t) + \gamma \max_a Q_\theta(s_{t+1}, a) - Q_\theta(s_t, a_t)].\end{aligned}$$

The results reported in Figure 3 and Figure 6 further confirm the theoretical predictions made via Definition 3.2 and Proposition 3.4. In addition to the results for individual games reported in Figure 3, we compute a normalized measure of the gain in temporal difference achieved when using MaxMin TD Learning exploration and plot the median across games. We define the normalized $\mathcal{TD}$ gain to be

$$\text{Normalized } \mathcal{TD} \text{ Gain} = 1 + \frac{\mathcal{TD}_{\text{method}} - \mathcal{TD}_{\epsilon\text{-greedy}}}{|\mathcal{TD}_{\epsilon\text{-greedy}}|}$$

where $\mathcal{TD}_{\text{method}}$ and $\mathcal{TD}_{\epsilon\text{-greedy}}$ are the temporal difference for any given exploration method and ϵ-greedy respectively. The leftmost and middle plot of Figure 6 report the median across all games of the normalized $\mathcal{TD}$ gain results for MaxMin TD Learning and NoisyNetworks in the Arcade



Median

80th Percentile

Figure 5: Human normalized scores median and 80th percentile over all games in the Arcade Learning Environment (ALE) 100K benchmark for MaxMin TD Learning algorithm and the canonical exploration algorithm ϵ-greedy for QRDQN.

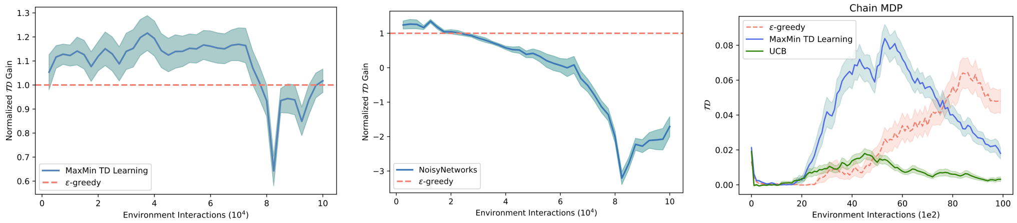


Figure 6: Left and Middle: Normalized temporal difference $\mathcal{TD}$ gain median across all games in the Arcade Learning Environment 100K benchmark for MaxMin TD Learning and NoisyNetworks. Right: Temporal difference $\mathcal{TD}$ when exploring chain MDP with Upper Confidence Bound (UCB) method, ϵ-greedy and our proposed algorithm MaxMin TD Learning.

Learning Environment 100K benchmark. Note that, consistent with the predictions of Proposition 3.4, the median normalized temporal difference gain for MaxMin TD Learning is up to 25 percent larger than that of ϵ-greedy. The results for NoisyNetworks demonstrate that alternate exploration methods lack this positive bias relative to the uniform random action.

The fact that, as demonstrated in Table 1, MaxMin TD Learning significantly outperforms noisy networks in the low-data regime is further evidence of the advantage the positive bias in temporal difference confers. The rightmost plot of Figure 6 reports $\mathcal{TD}$ for the motivating example of the chain MDP. As in the large-scale experiments, prior to convergence MaxMin TD Learning exhibits a notably larger temporal difference relative to the canonical baseline methods.

7 Conclusion

In our study we focus on the following questions in deep reinforcement learning: *(i) Is it possible to increase sample efficiency in deep reinforcement learning in a computationally efficient way with conceptually simple choices?, (ii) What is the theoretical motivation of our proposed perspective, simply minimizing the state-action value function in early training, that results in one of the most computationally efficient ways to explore in deep reinforcement learning?* and, *(iii) How would the theoretically motivated simple idea transfer to large scale experiments in high-dimensional state representation MDPs?* To be able to answer these questions we propose a novel, theoretically motivated method with zero additional computational cost based on following actions that minimize the state-action value function to explore in deep reinforcement learning. We demonstrate theoretically that our method MaxMin TD Learning based on minimization of the state-action value results in higher temporal difference, and thus creates novel transitions in exploration with more unique experience collection. Following the theoretical motivation we initially show in a toy example in the chain MDP setup that our proposed method MaxMin TD Learning results in achieving higher sample efficiency. Then, we expand this intuition and conduct large scale experiments in the Arcade Learning Environment, and demonstrate that our proposed method MaxMin TD Learning increases the performance on the Arcade Learning Environment 100K benchmark by 248%.

REFERENCES

Oron Anschel, Nir Baram, and Nahum Shimkin. Averaged-dqn: Variance reduction and stabilization for deep reinforcement learning. *International Conference on Machine Learning (ICML)*, 2017.

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.*, pp. 253–279, 2013.

Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML*, volume 70 of *Proceedings of Machine Learning Research*, pp. 449–458. PMLR, 2017.

Ronen I Brafman and Moshe Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 2002.

Will Dabney, Mark Rowland, Marc G. Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 2892–2901. AAAI Press, 2018.

Sebastian Flennerhag, Yannick Schroecker, Tom Zahavy, Hado van Hasselt, David Silver, and Satinder Singh. Bootstrapped meta-learning. *10th International Conference on Learning Representations, ICLR*, 2022.

Jessica Hamrick, Victor Bapst, Alvaro SanchezGonzalez, Tobias Pfaff, Theophane Weber, Lars Buesing, and Peter Battaglia. Combining q-learning and search with amortized value estimates. In *8th International Conference on Learning Representations, ICLR*, 2020.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. *Association for the Advancement of Artificial Intelligence (AAAI)*, 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 *Thirty-second AAAI conference on artificial intelligence*, 2018.

Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Mohammadamin Barekatain, Simon Schmitt, and David Silver. Learning and planning in complex action spaces. In *Proceedings of the 38th International Conference on Machine Learning, ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4476–4486. PMLR, 2021.

Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. *International Conference on Machine Learning (ICML)*, 2013.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 2002.

Qingfeng Lan, Yangchen Pan, Alona Fyshe, and Martha White. Maxmin q-learning: Controlling the estimation bias of q-learning. *International Conference on Learning Representations (ICLR)*, 2020.

Xiuyuan Lu and Benjamin Van Roy. Information-theoretic confidence bounds for reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 2458–2466, 2019.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, arc G Bellemare, Alex Graves, Martin Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518: 529–533, 2015.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016*, volume 48, pp. 1928–1937, 2016.

Junhyuk Oh, Matteo Hessel, Wojciech M. Czarnecki, Zhongwen Xu, Hado van Hasselt, Satinder Singh, and David Silver. Discovering reinforcement learning algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Tom Schaul, John Quan, Ioannis Antonogloua, and David Silver. Prioritized experience replay. *International Conference on Learning Representations (ICLR)*, 2016.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy P. Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588, 2020.

Hado van Hasselt. Double q-learning. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010.*, pp. 2613–2621. Curran Associates, Inc., 2010.

Hado van Hasselt, Matteo Hessel, and John Aslanides. When to use parametric models in reinforcement learning? In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems*, pp. 14322–14333, 2019.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando. De Freitas. Dueling network architectures for deep reinforcement learning. *Internation Conference on Machine Learning ICML.*, pp. 1995–2003, 2016.