Dead-ends and Secure Exploration in Reinforcement Learning
* Supplementary Material *

Mehdi Fatemi [1] **Shikhar Sharma** [1] **Harm van Seijen** [1] **Samira Ebrahimi Kahou** [2]

S1. Proof of Theorems

Theorem 1. *Let q_e^* be the optimal value function of $\mathcal{M}_e$ under P2.1 and P2.2. Let further η be any arbitrary policy that satisfies the following:*

$$\eta(s,a) \leq 1 + q_e^*(s,a), \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A} \qquad (1)$$

where $q_e^*(s,\cdot) \neq -1$ at least for one action. Then η is secure.

Proof. The two conditions jointly imply that the optimal value of *all* and *only* the dead-end states will be exactly -1 regardless of the length of dead-end trajectories. Specifically, under P1 and P2, direct evaluation of Bellman's equation follows that for all dead-end states $s' \in \mathcal{S}_D$ and all actions $a \in \mathcal{A}$, $q_e^*(s',a) = -1$. It directly implies $\max_{a'} q_e^*(s',a) = -1$. Additionally, it implies that if $q_e^*(s,\cdot) \neq -1$ at least for one action, then s is not a dead-end. For non-dead-end states $s \in \mathcal{S} \backslash \mathcal{S}_\mathcal{D}$ we therefore get:

$$\begin{aligned} q_e^*(s,a) &= -\sum_{s' \in \mathcal{S}_D} T(s,a,s') + \\ &\qquad \sum_{s' \notin \mathcal{S}_D} T(s,a,s') \max_{a'} q_e^*(s',a') \\ &= -\sum_{s' \in \mathcal{S}_D} T(s,a,s') - \beta(s,a), \end{aligned} \qquad (2)$$

where,

$$\beta(s,a) = -\sum_{s' \notin \mathcal{S}_D} T(s,a,s') \max_{a'} q_e^*(s',a') \geq 0$$

Of note, only $\beta \geq 0$ is required for the proof. However, we can tell more precisely that $\beta \in [0,1)$ because if $s' \notin \mathcal{S}_D$ then $\max_{a'} q_e^*(s',a') \in (-1,0]$. Also, s is not a dead-end; hence, $\sum_{s' \notin \mathcal{S}_D} T(s,a,s') \in (0,1]$.

Using the antecedent of Property 1 as well as $\beta \geq 0$, it therefore yields:

$$\begin{aligned} q_e^*(s,a) &\leq q_e^*(s,a) + \beta(s,a) \\ &= -\sum_{s' \in \mathcal{S}_D} T(s,a,s') \\ &\leq -(1-\lambda) \end{aligned}$$

which implies $1 + q_e^*(s,a) \leq \lambda$. Hence, setting the following

$$\mu(s,a) \leq 1 + q_e^*(s,a) \qquad (3)$$

will hold the consequent, thereby assuring Property 1, and μ is secure by definition. □

Theorem 2. *Under P2.1 and P2.2, let v_e^* and q_e^* be the optimal state and state-action value functions of $\mathcal{M}_e$. Then there exists a gap between $v_e^*(s')$ and $q_e^*(s,a)$ for all $a \in \mathcal{A} \backslash \mathcal{A}_D(s)$, $s' \in \mathcal{S}_D$, and $s \in \mathcal{S} \backslash \mathcal{S}_D$. Furthermore, the gap is independent of dead-end's possible length.*

Proof. Similarly to Theorem 1, the two conditions jointly imply that the optimal value of all the states on all dead-end trajectories will be exactly -1 regardless of their length. In particular, $v_e^*(s') = \max_{a'} q_e^*(s',a') = -1$, $\forall s' \in \mathcal{S}_D$, which implies $q_e^*(s,a) = -1$ if $\sum_{s' \in \mathcal{S}_D} T(s,a,s') = 1$. On the other hand, for all non-dead-end states s, $\min_{a'} q_e^*(s,a') > -1$ because there always exists at least one action that transitions to a non-dead-end state (due to the assumption of s being non-dead-end itself). Formally, for a non-dead-end state s, we have $T(s,a,s') < 1$, $\forall s' \in \mathcal{S}_D$, which implies $q_e^*(s,a) = -\sum_{s'} T(s,a,s') > -1$. As a result, there will be a theoretical gap between $q_e^*(s,a) \neq -1$ and $v_e^*(s') = -1$, which only depends on the transition probabilities $T(s,a,s')$ and not the length of dead-ends. □

Theorem 3. *If the following hold:*

1. *States and actions are finite (tabular settings).*
2. *Policy η exists that satisfies* Property 1*, and η is used as the sole behavioural policy. Furthermore,*

[1]Microsoft Research, 2000 McGill College Avenue, Suite 550, Montréal, QC H3A 3H3, Canada [2]McGill University, 845 Sherbrooke Street West, Montréal, QC H3A 0G4, Canada. Correspondence to: Mehdi Fatemi <mehdi.fatemi@microsoft.com>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

η visits all the non-dead-end *states infinitely often.*

3. *$q_\pi(\cdot,\cdot)$ is initialized pessimistically and standard conditions are applied on the step-size α_π.*

then, Q-learning of $q_\pi(s,a)$ converges to $q^*_\pi(s,a)$ for $s \in \mathcal{S}\backslash\mathcal{S}_D$ and $a \in \mathcal{A}\backslash\mathcal{A}_D(s)$.

Proof. Recall that $\mathcal{A}_D(s)$ denotes the set of all actions a_d, where $T(s,a_d,s') = 1$, $s' \in \mathcal{S}_D$. We first note that if there exists an action a at state $s \in \mathcal{S}\backslash\mathcal{S}_D$, which transitions to a dead-end with some probability less than 1 (hence $a \notin \mathcal{A}_D(s)$) then that specific dead-end and all the dead-end states after that on all possible trajectories will be seen infinitely often as s is also visited infinitely often by assumption. Therefore, the value of such dead-ends will be updated just as in standard Q-learning. Hence, this case does not affect convergence of $q_\pi(s,a)$ to its optimal value.

Now, let us focus on $a_d \in \mathcal{A}_D(s)$. Property 1 with $\lambda = 0$ brings

$$T(s,a_d,s') = 1 \implies \eta(s,a_d) = 0$$

which implies that at state s, η never selects $a_d \in \mathcal{A}_D(s)$; hence, the corresponding $q(s,a_d)$ values will remain at the initial value with no change. We therefore need to prove that never seeing transitions of the form (s,a_d,s'), $s \in \mathcal{S}\backslash\mathcal{S}_D$ and $s' \in \mathcal{S}_D$ does not affect convergence of $q_\pi(s,a)$ to $q^*_\pi(s,a)$ for $s \in \mathcal{S}\backslash\mathcal{S}_D$ and $a \in \mathcal{A}\backslash\mathcal{A}_D(s)$.

Consider the Q-learning update for a general transition from state s to s' with action a and reward r, namely

$$q(s,a) \leftarrow (1-\alpha)q(s,a) + \alpha\delta, \text{ with}$$
$$\delta = r + \gamma \max_{a' \in \mathcal{A}} q(s',a') \tag{4}$$

As explained, $q(s,a_d)$ remains at the initial value q_{init} for all $s \in \mathcal{S}\backslash\mathcal{S}_D$. Pessimistic initialization therefore implies that $q(s,a_d) = q_{init} \leq \max_{a' \in \mathcal{A}} q(s,a')$, $\forall a_d \in \mathcal{A}_D(s)$, which yields $\max_{a' \in \mathcal{A}} q(s,a') = \max_{a' \in \mathcal{A}\backslash\mathcal{A}_D(s)} q(s,a')$. Hence, the Q-learning target δ in (4) remains unchanged if $\mathcal{A}_D(s)$ is excluded from the behavioural policy.

We have so far established that transitioning to a dead-end state with non-zero probability and exclusion of forceful transitions to dead-ends from exploration (both of which are resulted from the assumptions of this theorem) do not affect Q-learning update of $q_\pi(s,a)$ for $s \in \mathcal{S}\backslash\mathcal{S}_D$ and $a \in \mathcal{A}\backslash\mathcal{A}_D(s)$. Finally, we note that Property 1 still maintains the probability of selecting other actions to be non-zero, which together with the second part of assumption 2 fulfills the general conditions under which Q-learning converges, and therefore Q-learning still converges to the optimal value for all the stipulated state-action pairs. As a final note, the value of excluded state and actions by this theorem will be remaining between, at minimum, the initial value q_{init} and, at maximum, their optimal value, depending how many updates are taken on each of those state-action pairs before the algorithm stops. □

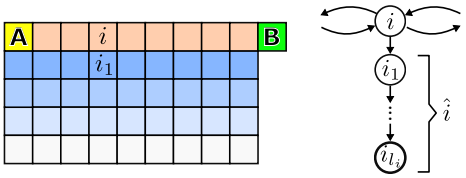

Figure 1. The bridge problem. Left: Agent starts from **A** and should reach **B**. Each vertical blue path is an inescapable and uncontrollable fall trajectory with a random length l_i (demonstrated by the gradient blue). Right: The corresponding part of MDP for state i on the bridge.

S2. The Bridge Game

Consider a bridge of length L, shown in Figure 1. The agent starts at the left end of the bridge and its goal is to reach the right end, resulting in some very large reward $r_B \gg +1$. At each step, n_a actions are available, where action a_n goes right with probability (w.p.) x_n, goes left w.p. y_n, and falls down w.p. z_n. Assume the following:

- For a_0: x_0 close to 1, $y_0 = 0$ and z_0 close to 0 ($z_0 = 1 - x_0$);
- For a_1: $x_1 = z_1 = 0$ and $y_1 = 1$;
- For all other actions, a_j, $j \geq 2$: $x_j = y_j = 0$, and $z_j = 1$.

Once the agent falls, it takes random (but bounded) number of uncontrollable steps to terminate and return to the initial position. Therefore, the agent is always at the risk of falling into a random-length inescapable trajectory before the episode *undesirably* terminates and it can restart.

There are two important possibilities for setting the reward, which we study separately:

(R1) $+r_B$ if reaching **B** and zero everywhere else, and

(R2) -1 if termination happens before reaching **B**, $+r_B$ if reach **B**, and zero otherwise.

In both cases, direct evaluation of Bellman equation yields the following result.

Proposition 1. *If r_B is sufficiently large and $\gamma < 1$, the optimal policy is $\pi(s,a_0) = 1$, for all the bridge states s.*

The case of R1 brings the following:

Proposition 2. *Under conditions of R1, a tabular agent with zero initialization (no prior) and with any one of ϵ-greedy (with or without annealing), fully greedy, fully random, or Boltzmann exploration will reach the goal for the first time with probability*

$$p = \big(\frac{1}{1-(x_0 y_1/n_a^2)}\big)\big(\frac{x_0}{n_a}\big)^L$$

Proof. Since there is no reward, all the q-values remain zero (or any initial value) until the goal is reached for the first time. As a result, the behavioural policy will be uniformly random regardless of the mentioned exploration techniques, which implies that the probability of going one step right or left at any bridge state will be $\alpha = x_0/n_a$ and $\beta = y_1/n_a$, respectively. Hence, the probability of reaching **B** in exactly L steps is α^L (the probability of reaching **B** in less than L steps is trivially zero). For the agent to reach **B** in exactly $L+1$ steps, it must go left in one and only one of the states on the bridge, and go right all other times. Say it selects a_1 at step $\tau < L$ and selects a_0 all over, before and after τ. The probability that it reaches **B** in exactly $L+1$ steps will then be $\alpha^\tau \beta \alpha^{L-\tau+1} = \beta\alpha^{L+1}$. More generally, the probability that the agent reaches **B** in exactly $L+M$ steps is $\beta^M \alpha^{L+M}$. It therefore concludes that the probability of reaching **B** in at most $L+M$ steps is the following:

$$p_M(L) = \sum_{k=0}^{M} \beta^k \alpha^{L+k} = \alpha^L \Big(\frac{1-(\alpha\beta)^{M+1}}{1-\alpha\beta}\Big), \;\; M \geq 0 \tag{5}$$

The probability of reaching **B** for the first time is obtained by taking the limit from (5) as $M \to +\infty$, which proves the proposition. □

For a long-enough bridge, therefore, the agent reaches **B** for the first time only with extremely small probability, which goes to zero as L grows (since $x_0 < 1$). Consequently, for the corresponding agent, Q-learning with any one of the mentioned exploration techniques will need painfully long time to learn the optimal values. Additionally, Proposition 1 explicitly refers to the first time of reaching **B**, and does not assume any specific learning method; hence, it equally applies to other methods than Q-learning.

In the case of R2, let us first study Q-learning with zero initialization and greedy behavioural policy.

Proposition 3. *For the bridge problem of R2, Q-learning with zero initialization and greedy behavioural policy will converge to $\pi(A, a_1) = 1$, with probability that goes to 1 as L goes to infinity.*

Proof. Let i and $\hat{i}$ denote the i-th position on the bridge and its corresponding fall trajectory. Additionally, let i_j for $j = 1, \cdots, l_i$ be the j-th state on $\hat{i}$, with l_i denoting $\hat{i}$'s random length. By definition, i_{l_i} is a terminal state and its value is zero. Starting from i_{l_i-1}, because the behavioural policy is greedy and all q values are initialized at zero, it takes exactly n_a *visits* until $max_{a'} q(i_{l_i-1},\ a')$ becomes negative (precisely -1 if step-size is 1). Hence, on average, it takes $\bar{l_i} n_a$ visits of $\hat{i}$ until i_1 assumes a negative value, with $\bar{l_i} = \mathbb{E} l_i$ (again, if step-size is 1, then the value is $-\gamma^{\bar{l_i}-2}$). Let i be the first state on the bridge, for which i_1 becomes negative. Then, the very first time that $(i,\ a_0,\ i_1)$ is observed, if the positive value of the goal state has not yet been propagate back to $i+1$, then $q(i, a_0)$ will become negative. On the other hand, $q(i, a_1)$ remains zero because $max_{a'} q(i-1, a') = 0$ due to the assumption that i_1 is the first one that assumes negative max value. Hence, i becomes a barrier and the greedy policy will never pass it. Continuing with the same line of argument, after sufficient number of steps another state $j < i$ will become a new barrier and the learning indeed converges to always taking a_1 at the initial state. Next, one can observe that the minimum number of steps to receive r_B is L. It implies the probability that positive values propagate back to i goes to zero as $L \to +\infty$ for all $i < L$. Hence, the probability that a barrier emerges goes to 1 as L grows, which concludes the proof. □

The proof highlights the emergence of *barriers*, which is an interesting phenomenon. Indeed, barriers are responsible for greedy actions to fail under R2.

Conversely, using a fully random behavioural policy has extremely small chance of reaching **B** for the first time as L becomes large, just similar to that of Proposition 1. As a result, methods that combine greedy behaviour with some perturbation, such as ϵ-greedy or Boltzmann, will land on getting stuck behind barriers (as in Proposition 3) or having to wait for extremely large number of steps to see r_B for the first time (as in Proposition 1) and yet more training time to converge.

Another strategy that is also important to be discussed is optimistic initialization, which in theory should converge faster than ϵ-greedy and is the basis for several other methods. However, optimistic initialization (and similar techniques) may not be useful in stochastic environments because the exploration vanishes quickly to zero; hence, there is no guarantee of convergence. Additionally, it presents an exhaustive search, which makes it challenging to scale.

The aforementioned bridge problem is not the only configuration that is severely problematic. In actual fact, one can experimentally show that for certain values of x, y, and z, Q-learning will simply never converge in reasonable time.

S3. On the *Insecurity* of Boltzmann Policy with Negative Rewards

As emphasized in the main text, assigning (large) negative rewards to undesired terminal states is not a general solution when function approximation is used. However, in tabular settings it should not be a problem, at least in principle. In this section, we show that even if negative rewards are used, the Boltzmann policy is still not secure. Here we consider Boltzmann policy with zero initialization.

Let us assume that positively rewarded transitions are relatively distant. Hence, we may assume that the dead-end values next to the initial state s converge before any positive value propagates back. This is a fair assumption if the bridge effect is severe. For simplicity, assume also that all the dead-ends have same length of L. Before the positive values have the chance of propagating back to the neighboring states of s, the value of all the neighbors are non-positive. The reward for any transition starting from (s, a) is also non-positive (by the assumption of positive rewards being distant). One can therefore write:

$$\begin{aligned} q(s,a) &\leq \sum_{s' \in \mathcal{S}_D} T(s,a,s')\gamma \max_{a'} q(s',a') + \\ &\qquad \sum_{s' \notin \mathcal{S}_D} T(s,a,s')\gamma \max_{a'} q(s',a') \\ &\leq \sum_{s' \in \mathcal{S}_D} T(s,a,s')\gamma(-\gamma^{L-1}) + \\ &\qquad \sum_{s' \notin \mathcal{S}_D} T(s,a,s')\gamma \max_{a'} q(s',a') \\ &\leq -\gamma^L \sum_{s' \in \mathcal{S}_D} T(s,a,s') - \beta, \end{aligned}$$

where $\beta \geq 0$. Hence, antecedent of Property 1 implies:

$$\begin{aligned} &\frac{q(s,a)+\beta}{-\gamma^L} \geq \sum_{s' \in \mathcal{S}_D} T(s,a,s') \geq 1-\lambda \\ \implies &q(s,a) \leq q(s,a) + \beta \leq -\gamma^L(1-\lambda) \\ \implies &\eta(s,a) \propto e^{q(s,a)} \leq e^{-\gamma^L(1-\lambda)} \end{aligned} \tag{6}$$

where the equality holds if for example $1-\lambda$ is the strict value (i.e., $\sum_{s' \in \mathcal{S}_D} T(s,a,s') = 1-\lambda$), and if there exists at least one deterministic transition from s to a zero-valued state and the reward of such a transition is zero, which is not unlikely to happen at all. Hence, the gap between $\eta(s,a)$ and $e^{-\gamma^L(1-\lambda)}$ can be arbitrarily small (possibly zero).

Over the interval $\lambda \in [0,1]$, one can show that $e^{-\gamma^L(1-\lambda)} \geq \lambda$, with the equality holds if $\lambda = 1$ and the maximum gap occurs at $\lambda = 0$ (the curve $y = e^{c(x-1)}$, $c > 0$, is always above the line $y = x$ over the interval $x \in [0,1]$ and they touch at $y = x = 1$). It implies that even after considering the normalization factor, there is no guarantee that $\eta(s,a)$ holds Property 1; hence, Boltzmann policy is not secure in general. In particular, the probability that η becomes insecure increases as L increase and/or γ decreases. More importantly, (6) asserts that the Boltzmann policy is insecure almost surely when λ becomes very small. Therefore, the actions that almost surely transition to a dead-end are still likely to be chosen. In practical situations, one can easily show that the probability of taking an action, which deterministicaly transitions to a dead-end, can be totally non-negligible. This makes the algorithm very sensitive to the chosen magnitude of negative rewards (or to the temperature parameter of Boltzmann distribution).

S4. Implementation Details

For the deep RL example, we use a deep neural network with experience replay and a very basic/simplistic prioritizing technique. For the prepossessing and main implementation details, we closely follow (Mnih et al., 2015) with a smaller network. For the Atari screen, we only used the red channel resized to 84x84 pixels, which is stacked with the history of three previous images to shape the state (a tensor of 4x84x84). The neural network consists of two convolutional layers with 16 filters of 8x8 and stride of 4, and 32 filters of 4x4 and stride of 2, respectively. The convolutional layers have ReLU activation and are followed by a dense layer of 256 ReLU units, which is connected to a dense linear layer that provides the action-values. We use RMSProp optimizer with learning rate 0.000125 and Huber loss similarly to (Mnih et al., 2015). We trained the exploration network for 2M transitions with fully secure random walk (i.e., $\epsilon = 1$), and then anneal ϵ to 0.1 in 1M more steps, just as usual DQN implementations. Both trainings start after at least 50000 transitions and when at least one rewarding transition is observed. The rest of hyper-parameters are the same as in (Mnih et al., 2015).

As described in the deep RL section of the paper, for the exploration network, we always clip the target value to remain in $[-1, 0]$. For both networks, we maintain a very small additional experience replay, which only keeps the *non-zero rewarding transitions* in a FIFO manner. Once sampling a minibatch, we replace *only one* of the samples (out of 32, which is the minibatch size) by one sample from the rewarding buffer with probability 1.0 for exploration and 0.1 for exploitation agents. The size of rewarding buffers are 20 and 200 transitions for the exploitation and exploration agents, respectively. Notice that the number of rewarding transitions are normally much larger for undesired terminations (hence, for the exploration agent). Finally, for the exploration network, a life-loss is considered as an undesired terminal state with $r_e = -1$, and zero reward elsewhere, according to P1 (see Section 2.2. of the main text). No other terminal state is considered for exploration. Additionally, a life-loss is not

necessarily the end of the game, and the environment may continue if more lives are still available. Note also that any life-loss is directly signalled from ALE, and there is no need to accessing the RAM.

References

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.