REVISITING PRIORITIZED EXPERIENCE REPLAY: A VALUE PERSPECTIVE

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) agents need to learn from past experiences. Prioritized experience replay that weighs experiences by the magnitude of their temporal-difference (TD) error significantly improves the learning efficiency for RL algorithms. The magnitude of TD error, also referred to as *surprise*, quantifies the unexpectedness of an experience to the learning agent. However, how *surprise* is related to the importance of experience is not well understood. To address this problem, we derive three value metrics to quantify the importance of experience, which consider the extra reward would be earned by accessing the experience. We theoretically show these value metrics are upper-bounded by *surprise* for Q-learning. Furthermore, we successfully extend our theoretical framework to maximum-entropy RL by deriving the upper bound of these value metrics for soft Q-learning, which is also related to *surprise*. Our framework links two important quantities in RL, *i.e.*, *surprise* and value of experience, and provides a theoretical basis to estimate the value of experience by *surprise*. We empirically show that the upper bounds hold in practice, and experience replay using the upper bound as priority improves maximum-entropy RL in Atari games.

1 INTRODUCTION

Learning from important experiences prevails in nature. Research of rodent hippocampus shows that memories with higher importance, such as those associated with high-rewarding locations or high reward-prediction errors, are replayed more frequently (Michon et al., 2019; Roscow et al., 2019; Salvetti et al., 2014). Research (Gruber et al., 2016; Schapiro et al., 2018) also shows that people who have more frequently replay of high-reward associated memories show better performance in memory tasks. A normative theory suggests that prioritized memory access according to the utility of memory can explain hippocampal replay across different tasks (Mattar & Daw, 2018). As accumulating new experiences is costly in reality, utilizing valuable past experiences is a key for efficient learning (Ólafsdóttir et al., 2018).

Differentiating important experiences from unimportant ones also benefits reinforcement learning (RL) (Katharopoulos & Fleuret, 2018). Prioritized experience replay (PER) (Schaul et al., 2016) is an experience replay technique built on deep Q-network (DQN) (Mnih et al., 2015). PER weighs samples by the magnitudes of their temporal-difference (TD) errors; *i.e.*, the experiences with higher magnitude of TD error are sampled more frequently. By prioritized sampling, PER significantly improves the learning efficiency and has been adopted (Hessel et al., 2018; Horgan et al., 2018; Kapturowski et al., 2019) and extended (Daley & Amato, 2019; Pan et al., 2018; Schlegel et al., 2019) by various deep RL algorithms . The magnitude of TD error, referred to as *surprise*, quantifies the unexpectedness of an experience to a learning agent. *Surprise* biologically corresponds to the signal of reward prediction error in dopamine system (Schultz et al., 1997; Glimcher, 2011), which affects memory of animal and human (Lisman & Grace, 2005; McNamara et al., 2014). However, how *surprise* is related to the importance of experience in the context of RL is not well understood.

To address this problem, we theoretically derive the relationship between *surprise* and value of experience. In our framework, we use three value metrics of accessing experiences to represent their importance. The expected value of backup (EVB) of experience is formalized, from the first principle, as the increase in future rewards resulting from such memory access (Mattar & Daw, 2018). According to policy iteration (Sutton & Barto, 2018), EVB is further separated into evaluation im-

provement value (EIV) and policy improvement value (PIV). We mathematically show that these value metrics are upper-bounded by *surprise*. Therefore, *surprise* implicitly tracks the value of experience, and accounts for the importance of experience. We further extend our framework to maximum-entropy RL, which augments the reward with an entropy term to encourage exploration (Haarnoja et al., 2017). We derive the upper bound for these value metrics for soft Q-learning, which is also related to *surprise*. Experiments in Maze and CartPole support our theoretical results for both tabular and function approximation RL methods, showing that the derived upper bounds hold tightly in practice. Moreover, we also show that experience replay using the upper bound as priority improves maximum-entropy RL (*i.e.*, soft DQN) in Atari games.

Our framework links two important quantities in RL, TD error and value of experience (Schultz et al., 1997; Mattar & Daw, 2018). Since the value of experience is typically hard to compute, our framework provides a theoretical basis to estimate it by *surprise*. Moreover, our framework can be potentially applied to other RL algorithms and help to prioritize experiences in a more effective way, *e.g.*, batch RL (Lange et al., 2012; Fu et al., 2020). Our study may also provide insights to memory deficit related to traumatic experience, *i.e.*, PTSD, with which people have abnormal attention to *surprise* (Brown et al., 2018).

2 Model

2.1 Markov Decision Process and Policy Iteration

We consider a Markov Decision Process (MDP) defined by a tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma\}$, where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions, $\mathcal{P}$ is the transition function, $\mathcal{R}$ is the reward function, and $\gamma \in [0, 1]$ is the discount factor. A policy π of an agent assigns probability $\pi(a|s)$ to each action $a \in \mathcal{A}$ given state $s \in \mathcal{S}$. The goal is to learn an optimal policy that maximizes the expected discounted return starting from time step t, $G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + ... = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$, where r_t is the reward the agent receives at time step t. Value function $v_\pi(s)$ is defined as the expected return starting from state s following policy π, and Q-function $q_\pi(s, a)$ is the expected return on performing action a in state s and subsequently following policy π. According to Q-learning (Watkins & Dayan, 1992), the optimal policy can be learned through policy iteration: performing policy evaluation and policy improvement interactively and iteratively. For each policy evaluation, we update $Q(s, a)$, an estimate of $q_\pi(s, a)$, by

$$Q_{\text{new}}(s, a) = Q_{\text{old}}(s, a) + \alpha \text{TD}(s, a, r, s'),$$

where TD error $\text{TD}(s, a, r, s') = r + \gamma \max_{a'} Q_{\text{old}}(s', a') - Q_{\text{old}}(s, a)$ and α is the step-size parameter. Q_{new} and Q_{old} denote the estimated Q-function before and after the update respectively. For each policy improvement, we update the policy from π_{old} to π_{new} according to the newly estimated Q-function,

$$\pi_{\text{new}} = \arg\max_a Q_{\text{new}}(s, a).$$

2.2 Value Metrics of Experience

The experience that the agent collected at time step k is represented by a 4-tuple $e_k = \{s_k, a_k, r_k, s'_k\}$. The utility of update on experience e_k is defined as the value added to the cumulative discounted rewards starting from state s_k, after updating on e_k. We denote such utility as the expected value of backup $\text{EVB}(e_k)$ (Mattar & Daw, 2018),

$$\begin{aligned} \text{EVB}(e_k) &= v_{\pi_{\text{new}}}(s_k) - v_{\pi_{\text{old}}}(s_k) \\ &= \sum_a \pi_{\text{new}}(a|s_k) q_{\pi_{\text{new}}}(s_k, a) - \sum_a \pi_{\text{old}}(a|s_k) q_{\pi_{\text{old}}}(s_k, a) \end{aligned} \tag{1}$$

where π_{old}, $v_{\pi_{\text{old}}}$ and $q_{\pi_{\text{old}}}$ are respectively the policy, value function and Q-function before the update, and π_{new}, $v_{\pi_{\text{new}}}$, and $q_{\pi_{\text{new}}}$ are those after the update. As the update on experience e_k consists of policy evaluation and policy improvement, the value of experience can further be separated into evaluation improvement value $\text{EIV}(e_k)$ and policy improvement value $\text{PIV}(e_k)$ by rewriting (1):

$$\text{EVB}(e_k) = \underbrace{\sum_a [\pi_{\text{new}}(a|s_k) - \pi_{\text{old}}(a|s_k)] q_{\pi_{\text{new}}}(s_k, a)}_{\text{PIV}(e_k)} + \underbrace{\sum_a \pi_{\text{old}}(a|s_k) [q_{\pi_{\text{new}}}(s_k, a) - q_{\pi_{\text{old}}}(s_k, a)]}_{\text{EIV}(e_k)}, \tag{2}$$

where PIV(e_k) measures the improvement due to the change of the policy, and EIV(e_k) captures the improvement on the estimate of Q-function. Thus, we have three metrics for the value of experience.

2.3 Value Metrics of Experience in Q-Learning

For Q-learning, we use Q-function to estimate the true action-value function. A backup over an experience e_k consists of policy evaluation with Bellman operator and greedy policy improvement. As the policy improvement is greedy, we can rewrite value metrics of experience to a simpler form. EVB can be written as follows from (1),

$$\text{EVB}(e_k) = \max_a Q_{\text{new}}(s_k, a) - \max_a Q_{\text{old}}(s_k, a). \tag{3}$$

From (2), PIV can be written as

$$\text{PIV}(e_k) = \max_a Q_{\text{new}}(s_k, a) - Q_{\text{new}}(s_k, a_{\text{old}}), \tag{4}$$

where $a_{\text{old}} = \arg\max_a Q_{\text{old}}(s_k, a)$, and EIV can be written as

$$\text{EIV}(e_k) = Q_{\text{new}}(s_k, a_{\text{old}}) - Q_{\text{old}}(s_k, a_{\text{old}}). \tag{5}$$

It is useful to extend the value metrics to function approximation methods, with which we learn a parameterized Q-function $Q(s, a; \theta_t)$. An update on experience e_k for parameters θ_t is,

$$\theta_{t+1} = \theta_t + \alpha(Q_{\text{target}}(s_k, a_k) - Q(s_k, a_k; \theta_t))\nabla_{\theta_t} Q(s_k, a_k; \theta_t)),$$

where α is the learning rate and Q_{target} is the target Q value, defined as

$$Q_{\text{target}}(s_k, a_k) = r_k + \gamma \max_{a'} Q(s'_k, a'; \theta_t).$$

As α in function approximation Q-learning is usually very small, for each update, the parameterized function moves to its target value only by a small amount. The value of an experience is fully revealed only if the function converges to the target value. For function approximation methods, we define the expected value of an experience (EVB) as the value added by updating on that experience as if the parameterized Q-function converges to its target value. Now, EVB, PIV and EIV can be written as:

$$\begin{aligned}
\text{EVB}(e_k) &= \max_a Q_{\text{target}}(s_k, a) - \max_a Q(s_k, a; \theta_t) \\
\text{PIV}(e_k) &= \max_a Q_{\text{target}}(s_k, a) - Q_{\text{target}}(s_k, a_{\text{old}}) \\
\text{EIV}(e_k) &= Q_{\text{target}}(s_k, a_{\text{old}}) - Q(s_k, a_{\text{old}}; \theta_t).
\end{aligned}$$

3 Upper Bounds for Value Metrics in Q-Learning

PER (Schaul et al., 2016) greatly improves the learning efficiency of DQN. However, the underlying rationale is not well understood. Here, we find in Q-learning, *surprise* is an upper bound for all the value metrics.

Theorem 3.1. *The three value metrics of experience e_k, i.e., EVB, PIV and EIV, in Q-learning are all bounded by surprise $|TD(s_k, a_k, r_k, s'_k)|$.*

Proof. The absolute EVB can be written as

$$\begin{aligned}
|\text{EVB}(e_k)| &= |\max_a Q_{\text{new}}(s, a) - \max_a Q_{\text{old}}(s, a)| \\
&\leq \max_a |Q_{\text{new}}(s, a) - Q_{\text{old}}(s, a)| \\
&\leq \alpha |\text{TD}(s_k, a_k, r_k, s'_k)|,
\end{aligned} \tag{6}$$

where the second line is from the contraction of max operator.

The absolute PIV can be written as

$$\begin{aligned}
|\text{PIV}(e_k)| &= |\max_a Q_{\text{new}}(s, a) - Q_{\text{new}}(s, \arg\max_a Q_{\text{old}}(s, a))| \\
&= \max_a Q_{\text{new}}(s, a) - Q_{\text{new}}(s, \arg\max_a Q_{\text{old}}(s, a)) \\
&\leq \alpha |\text{TD}(s_k, a_k, r_k, s'_k)|,
\end{aligned} \tag{7}$$

where the second line is from that the change in Q-function following greedy policy improvement is greater or equal to 0, and the third line is because the increase in Q-function resulted from greedy policy improvement will not exceeds the *surprise* (times the learning step-size). Similarly, the absolute EIV can be written as follows,

$$\begin{aligned}|\text{EIV}(e_k)| &= |Q_{\text{new}}(s, a_{\text{old}}) - Q_{\text{old}}(s, a_{\text{old}})| \\ &= \mathbf{1}_{\text{s}=\text{s}_\text{k},\text{a}_\text{old}=\text{a}_\text{k}} \alpha |\text{TD}(s_k, a_k, r_k, s'_k)| \\ &\leq \alpha |\text{TD}(s_k, a_k, r_k, s'_k)| \end{aligned} \tag{8}$$

For equations (6) and (7), the equality is reached if the action in the experience is the same as the best action after update. For (8), the equality is met if the action in the experience is the best action before update. □

In Theorem 3.1, we prove that |EVB|, |PIV|, and |EIV| are upper-bounded by the *surprise* (scaled by the learning step-size) in Q-learning. As *surprise* intrinsically tracks the prediction improvement as well as policy improvement, it can serve as an appropriate importance metric for past experiences. We will further study these relationship in experiments.

4 Extension to Maximum-Entropy RL

In this section, we extend our framework to study the relationship between *surprise* and value of experience in maximum-entropy RL, particularly, soft Q-learning.

4.1 Soft Q-Learning

Unlike regular RL algorithms, maximum-entropy RL augments the reward with an entropy term,

$$r + \beta \mathcal{H}(\pi(\cdot|s)),$$

where $\mathcal{H}(\cdot)$ is the entropy, and β is an optional temperature parameter that determines the relative importance of entropy and reward. The goal of maximum-entropy RL is to maximize the expected cumulative entropy-augmented rewards. Maximum-entropy RL algorithms have certain advantages over regular RL algorithms, such as capturing multiple modes of near optimal policies and better exploration and transfer between tasks.

Soft Q-learning is an off-policy value-based algorithm built on maximum-entropy RL principles (Haarnoja et al., 2017; Schulman et al., 2017). For soft Q-learning, the policy is stochastic. During policy iteration, Q-function is updated through soft Bellman operator Γ^{soft} and the policy is updated to a maximum-entropy policy:

$$\begin{aligned}&\text{Policy Evaluation: } Q^{\text{soft}}_{\text{new}}(s, a) = [\Gamma^{\text{soft}} Q^{\text{soft}}_{\text{old}}](s, a) = r + \gamma V^{\text{soft}}_{\text{old}}(s') \\ &\text{Policy Improvement: } \pi_{\text{new}}(a|s) = \text{softmax}_a(\frac{1}{\beta} Q^{\text{soft}}_{\text{new}}(s, a)),\end{aligned}$$

where $\text{softmax}_i(x) = \exp(x_i)/\sum_i \exp(x_i)$ is the softmax function, and the value function $V^{\text{soft}}_\pi(s)$ is defined as,

$$V^{\text{soft}}_\pi(s) = \mathbb{E}_a\{Q^{\text{soft}}_\pi(s, a) - \log(\pi(a|s))\} = \beta \log \sum_a \exp(\frac{1}{\beta} Q^{\text{soft}}_\pi(s, a)).$$

Similar as in Q-learning, the TD-error in soft Q-learning (soft TD error) is given by:

$$\text{TD}^{\text{soft}}(s, a, r, s') = r + \gamma V^{\text{soft}}_{\text{old}}(s') - Q^{\text{soft}}_{\text{old}}(s, a).$$

4.2 Value Metrics of Experience in Maximum-Entropy RL

Here, we extend the value metrics of experience to soft Q-learning. Expected value of backup for maximum-entropy RL (EVB^{soft}) is defined as,

$$\begin{aligned}\text{EVB}^{\text{soft}}(e_k) &= v^{\text{soft}}_{\text{new}}(s_k) - v^{\text{soft}}_{\text{old}}(s_k) \\ &= \sum_a \pi_{\text{new}}(a|s)\{q^{\text{soft}}_{\text{new}}(s,a) - \beta\log(\pi_{\text{new}}(a|s))\} \\ &\qquad - \sum_a \pi_{\text{old}}(a|s)\{q^{\text{soft}}_{\text{old}}(s,a) - \beta\log(\pi_{\text{old}}(a|s))\} \end{aligned} \tag{9}$$

EVB^{soft} can be separated into PIV^{soft} and EIV^{soft},

$$\begin{aligned}\text{PIV}^{\text{soft}}(e_k) &= \sum_a \pi_{\text{new}}(a|s)\{q^{\text{soft}}_{\text{new}}(s,a) - \beta\log(\pi_{\text{new}}(a|s))\} \\ &\qquad - \sum_a \pi_{\text{old}}(a|s)\{q^{\text{soft}}_{\text{new}}(s,a) - \beta\log(\pi_{\text{old}}(a|s))\} \\ &= \sum_a \{\pi_{\text{new}}(a|s) - \pi_{\text{old}}(a|s)\}q^{\text{soft}}_{\text{new}}(s,a) + \beta(H(\pi_{\text{new}}(\cdot|s)) - H(\pi_{\text{old}}(\cdot|s))), \end{aligned} \tag{10}$$

$$\text{EIV}^{\text{soft}}(e_k) = \sum_a \pi_{\text{old}}(a|s)[q^{\text{soft}}_{\text{new}}(s,a) - q^{\text{soft}}_{\text{old}}(s,a)]. \tag{11}$$

All value metrics of experience in maximum-entropy RL have similar forms as in regular RL except for the entropy term. This is because in maximum-entropy RL, the change of a policy also leads to the change of the policy entropy.

4.3 Upper Bounds for Value Metrics in Soft Q-learning

We theoretically derive upper bounds for the value metrics of experience in soft Q-learning.

Theorem 4.1. *The three value metrics of experience e_k, i.e., EVB, PIV and EIV, in soft Q-learning are bounded by $\rho_\pi * \left|TD^{soft}\right|$, where $\rho_\pi = \max\{\pi_{old}(a_k|s_k), \pi_{new}(a_k|s_k)\}$ is a policy related term.*

Proof. See Appendix A.1. □

The upper bound includes a policy term with the *surprise* (the magnitude of the soft TD error). The policy related term ρ_π represents the "on-policyness" of the experienced action before and after the update. Surprisingly, the coefficient of the entropy term β impacts the bound only through the policy term, which makes it an excellent priority even β changes during learning (Haarnoja et al., 2018). As $0 \leq \rho_\pi \leq 1$, the value metrics are also bounded by *surprise* $\left|\text{TD}^{\text{soft}}\right|$ alone, the same as in Q-learning. However, as $\pi(a_k|s_k)$ is usually less than 1, *surprise* is a looser upper bound for soft Q-learning. This supports previous study, which empirically shows that directly applying PER using *surprise* alone in maximum-entropy RL does not significantly improve the sample efficiency (Wang & Ross, 2019).

5 Experiments

Our experiments aim to answer following questions: (*i*) Does the theoretical upper bound for value metrics of experience hold true in practice? (*ii*) If it holds, is the upper bound tight? (*iii*) Does the upper bound derived for maximum-entropy RL help to improve performance? First, we implement tabular versions of Q-learning and soft Q-learning in Maze to verify the bound for tabular methods. Second, we implement DQN and soft DQN in CartPole to examine the bound for function approximation methods. Finally, we implement PER with the derived upper bound as priority for soft DQN and evaluate its effectiveness for maximum-entropy RL. The experimental details are described in Appendix A.2 and all the codes are available at: `https://github.com/RLforlife/VER`.

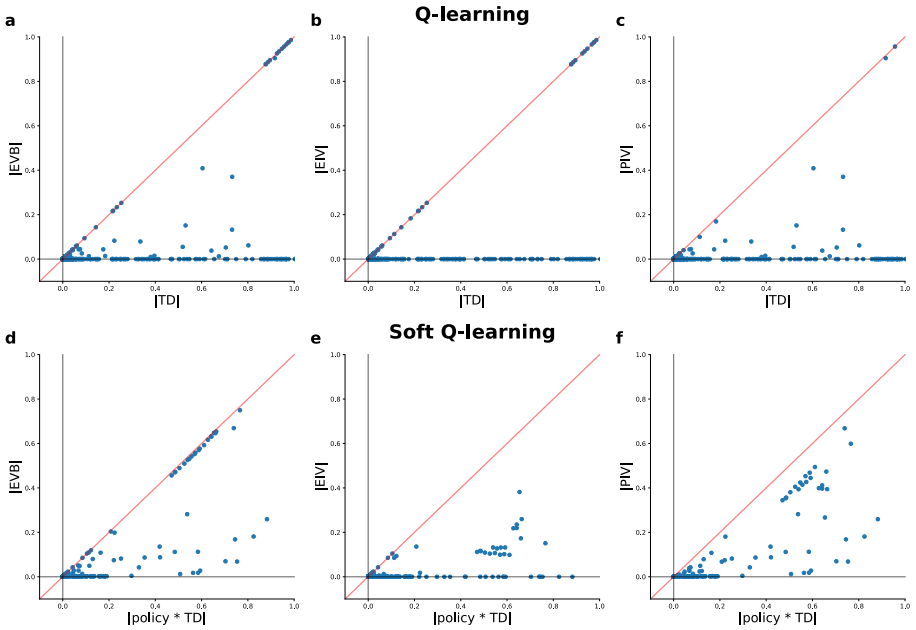

Figure 1: Results of Q-learning and soft Q-learning in Maze. **a-c.** Absolute value of EVB, EIV and PIV *v.s.* *surprise* (absolute value of TD error) in Q-learning. **d-f.** Absolute value of EVB, EIV and PIV *v.s.* the product of the policy term and *surprise* in soft Q-learning. The red line is the theoretical upper bound (the identity line).

5.1 MAZE

The first set of experiments are conducted in a maze environment of a 5×5 square with walls. The agent needs to reach the goal zone by moving one square in any of the four directions (north, south, east, west) each time. We implement tabular versions of Q-learning and soft Q-learning to solve this problem. The value metrics of an experience (EVB, PIV and EIV) for Q-learning is calculated using (3), (4), and (5). And those for soft Q-learning is calculated using (9), (10), and (11). For soft Q-learning, we do not model the actor explicitly: the policy is calculated as the softmax of the soft Q-function (see Section 4.1).

For each algorithm, the value metrics of experience as well as the theoretical upper-bound are illustrated in Figure 1. As we can see from upper panel, all three value metrics of experience are bounded by the *surprise* for Q-learning. As our theory predicts, the absolute values of EIV are either equal to the *surprise* (if the action of the experience is the best action before update) or 0. For soft Q-learning, as our theory predicts, all three value metrics of experience are bounded by the product of the *surprise* and the policy term. There is a large proportion of EVBs lies on the identity line, indicating the upper-bound is tight. The proportion of non-zero values of experiences is higher in soft Q-learning than in Q-learning. This is because all value metrics of experiences are affected by the "on-policyness" of the actions of experiences. Q-learning learns a deterministic policy that makes most actions of experiences off-policy, while soft Q-learning learns a stochastic policy that results in less sparse values of experiences.

5.2 CARTPOLE

CartPole is a pendulum with a center of gravity above its pivot point. The goal is to keep the pole balanced by moving the cart forward and backward. We implement DQN and soft DQN (DQN with soft-update) in this environment. For DQN, we implement the model according to Mnih et al. (2015), where we replace the Q-network with a two-layer MLP. For soft-DQN, all settings are the same with DQN, except for two modifications: during policy evaluation, the (soft) Q-network is updated according to the soft TD error; during policy improvement, the policy is updated following a maximum-entropy policy, as the softmax of soft Q values (see Section 4.1).

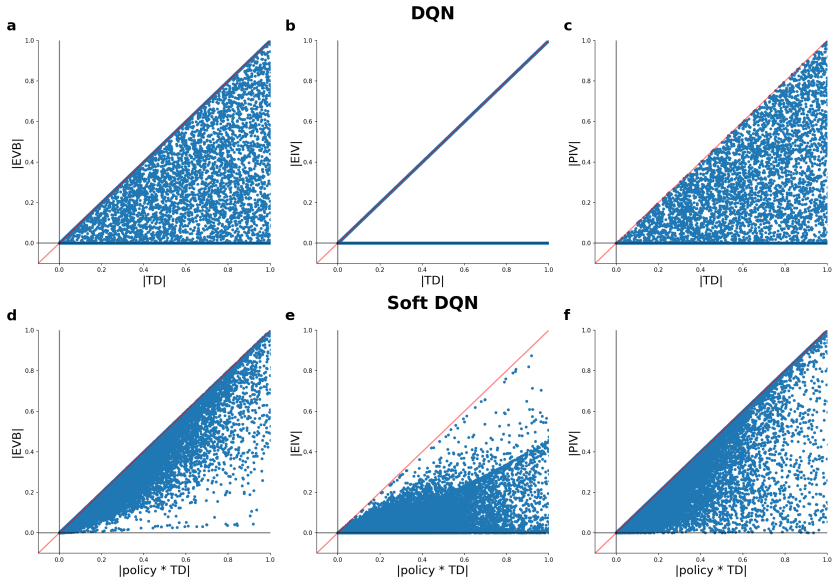

Figure 2: Results of DQN and soft DQN in CartPole. **a-c.** Absolute value of EVB, EIV and PIV *v.s. surprise* (the magnitude of the TD error) in DQN. **d-f.** Absolute value of EVB, EIV and PIV *v.s.* absolute value of the product of a policy term and the *surprise* (the magnitude soft TD error) in soft DQN. The red line is the theoretical upper bound (the identity line).

As depicted in Figure 2, all value metrics of experience in DQN and soft DQN are bounded by the theoretical upper bound. For DQN, absolute EVBs and PIVs are uniformly distributed in the bounded area, while absolute EIVs are equal to the *surprise* or 0 as in the tabular case. Results are different in soft DQN, where absolute EVBs and PIVs are distributed more closely towards the theoretical upper bounds, suggesting the upper bound in soft Q-learning is tighter. However, EIVs are distributed more closely to 0. This is because EIV represents evaluation improvement value of the experience, which does not consider policy improvement (see Appendix A.1). The experimental results confirm the theoretical predictions of the upper bounds for function approximation methods.

5.3 Atari Games

In this set of experiments, we investigate whether the theoretical upper bound for value metrics of experience can serve as an appropriate priority for experience replay in soft Q-learning. More specifically, we compare the performance of vanilla soft DQN and soft-DQN with PER, where we use *surprise* and the derived upper bound as priorities (Schaul et al., 2016), denoted as PER and VER (valuable experience replay) respectively. This set of experiments consists of nine games from Atari 2600 games. The goal is learning to play each of the games with the screen pixels as the only input. We closely follow the experimental setting and network architecture outlined by Mnih et al. (2015). For each game, the network is trained on a single GPU for 40M frames, or approximately 2 days.

Figure 3 shows that PER and VER have a higher learning speed than vanilla soft DQN in most of the games. On average, soft DQN with PER or VER outperform vanilla soft DQN by 11.8% or 18.0% respectively. Moreover, VER outperforms PER in most of the games (8.47% on average), which suggest that a tighter upper bound on value metrics indeed improves the performance of experience replay. These results suggest that the theoretical upper bound can serve as an appropriate priority for experience replay in soft Q-learning.

6 Discussion

In the paper, we formulate a framework to study the importance of experience for prioritized experience replay, and derive three value metrics of experience: expected value of backup, policy

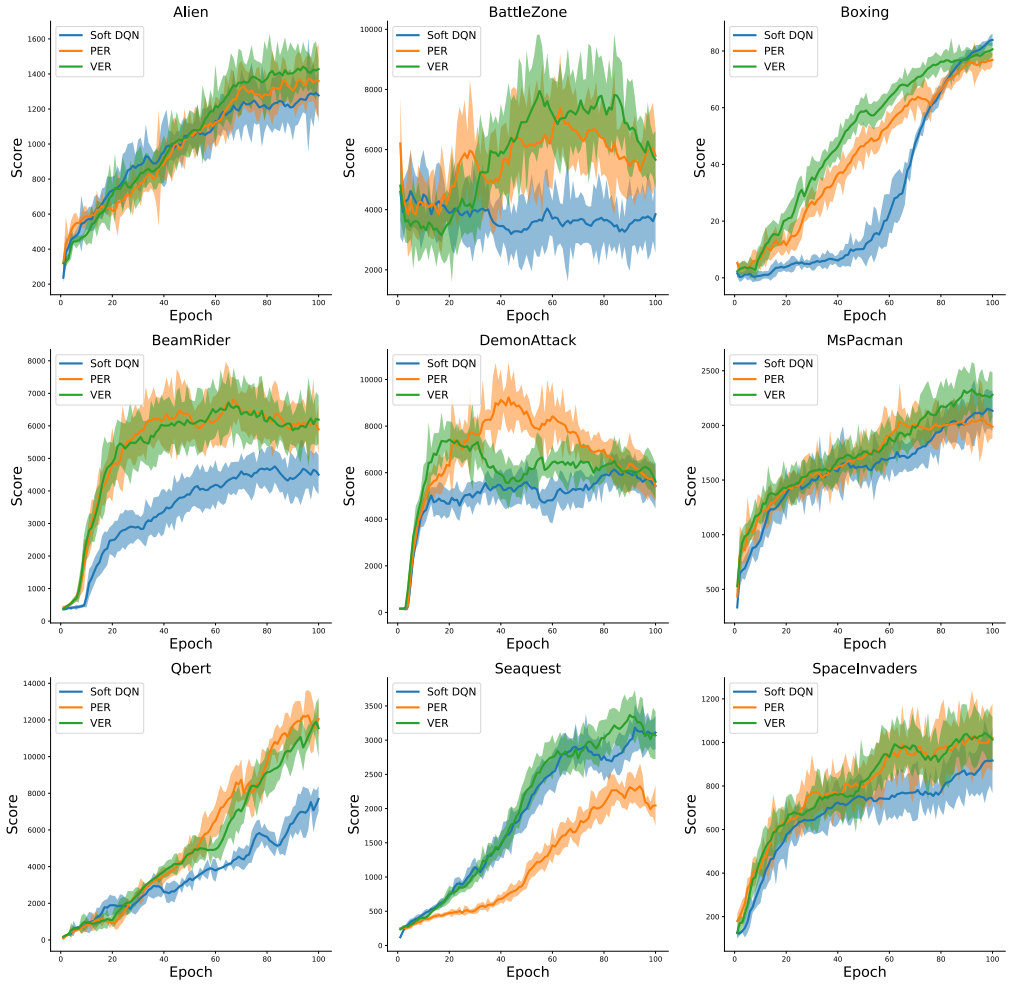

Figure 3: Learning curve of soft DQN (blue lines), and soft DQN with prioritized experience replay in term of soft TD error (PER, orange lines) and the theoretical upper bound for value metrics of experience (VER, green lines) on nine Atari games.

evaluation value, and policy improvement value. For Q-learning, we theoretically show these value metrics are upper bounded by *surprise* (the magnitude of the TD error), and the bound is tight when the action of experience is the same as the best action after update. Our claims are supported by the experiments of tabular Q-learning and DQN. Thus, *surprise* implicitly tracks the value of the experience, which leads to high sample efficiency of PER. Furthermore, we extend our framework to maximum-entropy RL, by showing that these value metrics are bounded by the product of a policy term and *surprise*. The results in soft Q-learning and soft DQN supports our theory. Moreover, we employ the upper bound as the priority for experience relay, termed as VER, in soft DQN. And we empirically show that VER outperforms PER and significantly improves the sample efficiency of soft DQN.

By linking *i.e. surprise* and value of experience, two important quantities in learning, our study has the following implications. First, from a machine learning perspective, our framework can help to establish new priorities for experience relay for different algorithms, *e.g.,* batch RL (Fu et al., 2020). Second, for neuroscience field, our work provides insight to how brain might encode the importance of experience. *Surprise* biologically corresponds to the reward prediction-error signal in the dopaminergic system (Schultz et al., 1997; Glimcher, 2011). Since *surprise* implicitly tracks the value of the experience, the brain may account on it to differentiate important experience. Third, our study may also provide insights to memory deficit related to traumatic experience, *i.e.*, PTSD, with which people have abnormal attention to *surprise* (Brown et al., 2018).

REFERENCES

Vanessa M Brown, Lusha Zhu, John M Wang, B Christopher Frueh, Brooks King-Casas, and Pearl H Chiu. Associability-modulated loss learning is increased in posttraumatic stress disorder. *eLife*, 7:e30150, 2018.

Brett Daley and Christopher Amato. Reconciling λ-returns with experience replay. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Paul W Glimcher. Understanding dopamine and reinforcement learning: the dopamine reward prediction error hypothesis. *Proceedings of the National Academy of Sciences*, 108(Supplement 3): 15647–15654, 2011.

Matthias J. Gruber, Maureen Ritchey, Shao-fang Wang, Manoj K. Doss, and Charan Ranganath. Post-learning Hippocampal Dynamics Promote Preferential Retention of Rewarding Events. *Neuron*, 89(5):1110–1120, 2016.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning (ICML)*, 2017.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

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 (AAAI)*, 2018.

Dan Horgan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado Van Hasselt, and David Silver. Distributed prioritized experience replay. In *International Conference on Learning Representations (ICLR)*, 2018.

Steven Kapturowski, Georg Ostrovski, John Quan, Remi Munos, and Will Dabney. Recurrent experience replay in distributed reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2019.

Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. *arXiv preprint arXiv:1803.00942*, 2018.

Sascha Lange, Thomas Gabel, and Martin Riedmiller. Batch reinforcement learning. In *Reinforcement learning*, pp. 45–73. Springer, 2012.

John E Lisman and Anthony A Grace. The hippocampal-vta loop: controlling the entry of information into long-term memory. *Neuron*, 46(5):703–713, 2005.

Marcelo G. Mattar and Nathaniel D. Daw. Prioritized memory access explains planning and hippocampal replay. *Nature Neuroscience*, 21(11):1609–1617, 2018.

Colin G McNamara, Álvaro Tejero-Cantero, Stéphanie Trouche, Natalia Campo-Urriza, and David Dupret. Dopaminergic neurons promote hippocampal reactivation and spatial memory persistence. *Nature neuroscience*, 17(12):1658–1660, 2014.

Frédéric Michon, Jyh-Jang Sun, Chae Young Kim, Davide Ciliberti, and Fabian Kloosterman. Post-learning hippocampal replay selectively reinforces spatial memory for highly rewarded locations. *Current Biology*, 29(9):1436–1444, 2019.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

H Freyja Ólafsdóttir, Daniel Bush, and Caswell Barry. The role of hippocampal replay in memory and planning. *Current Biology*, 28(1):R37–R50, 2018.

Yangchen Pan, Muhammad Zaheer, Adam White, Andrew Patterson, and Martha White. Organizing experience: a deeper look at replay mechanisms for sample-based planning in continuous state domains. *arXiv preprint arXiv:1806.04624*, 2018.

Emma L Roscow, Matthew W Jones, and Nathan F Lepora. Behavioural and computational evidence for memory consolidation biased by reward-prediction errors. *bioRxiv*, pp. 716290, 2019.

Beatrice Salvetti, Richard GM Morris, and Szu-Han Wang. The role of rewarding and novel events in facilitating memory persistence in a separate spatial memory task. *Learning & memory*, 21(2): 61–72, 2014.

Anna C Schapiro, Elizabeth A McDevitt, Timothy T Rogers, Sara C Mednick, and Kenneth A Norman. Human hippocampal replay during rest prioritizes weakly learned information and predicts memory performance. *Nature communications*, 9(1):1–11, 2018.

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

Matthew Schlegel, Wesley Chung, Daniel Graves, Jian Qian, and Martha White. Importance resampling for off-policy prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

John Schulman, Xi Chen, and Pieter Abbeel. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*, 2017.

Wolfram Schultz, Peter Dayan, and P Read Montague. A neural substrate of prediction and reward. *Science*, 275(5306):1593–1599, 1997.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Che Wang and Keith Ross. Boosting soft actor-critic: Emphasizing recent experience without forgetting the past. *arXiv preprint arXiv:1906.04009*, 2019.

Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3-4):279–292, 1992.

A Appendix

A.1 Proof of Theorem 4.1

For soft-Q learning, $|\text{EVB}^{\text{soft}}|$ can be written as

$$\begin{aligned}|\text{EVB}^{\text{soft}}(e_k)| &= |v^{\text{soft}}_{\text{new}}(s_k) - v^{\text{soft}}_{\text{old}}(s_k)| \\ &= |\beta \log \sum_a \exp(\frac{1}{\beta} Q^{\text{soft}}_{\text{new}}(s,a)) - \beta \log \sum_a \exp(\frac{1}{\beta} Q^{\text{soft}}_{\text{old}}(s,a))| \\ &= |\beta \log \sum_a \exp\left(\frac{1}{\beta}(Q^{\text{soft}}_{\text{old}}(s,a) + \mathbf{1}_{\text{a=a}_\text{k}} \text{TD}^{\text{soft}})\right) - \beta \log \sum_a \exp \frac{1}{\beta} Q^{\text{soft}}_{\text{old}}(s,a)|.\end{aligned}$$

Let us define the LogSumExp function $F(\vec{x}) = \beta \log \sum \exp\left(\frac{x_i}{\beta}\right)$. The LogSumExp function $F(\vec{x})$ is convex, and is strictly and monotonically increasing everywhere in its domain (El Ghaoui, Laurent 2017). The partial derivative of $F(\vec{x})$ is a softmax function

$$\frac{\partial F(\vec{x})}{\partial x_i} = \text{softmax}_i(\frac{1}{\beta}\vec{x}) \geq 0,$$

which takes the same form as the policy of soft Q-learning. For $\epsilon < 0$, we have:

$$\epsilon \frac{\partial F(x_1, ..., x_i, ...)}{\partial x_i} \leq F(x_1, ..., x_i + \epsilon, ...) - F(x_1, ..., x_i, ...) \leq 0.$$

Similarly, for $\epsilon \geq 0$, we have,

$$0 \leq F(x_1, ..., x_i + \epsilon, ...) - F(x_1, ..., x_i, ...) \leq \epsilon \frac{\partial F(x_1, ..., x_i + \epsilon, ...)}{\partial x_i}.$$

By substituting x_i by $Q^{\text{soft}}_{\text{old}}(s, a_i)$ and ϵ by TD^{soft}, and rewriting partial derivative of $F(\vec{x})$ into policy form, we have following inequalities. For $\text{TD}^{\text{soft}} \leq 0$

$$\pi_{\text{old}}(a_k|s_k)\text{TD}^{\text{soft}} \leq \beta \log \sum_a \exp(\frac{1}{\beta} Q^{\text{soft}}_{\text{new}}(s, a)) - \beta \log \sum_a \exp(\frac{1}{\beta} Q^{\text{soft}}_{\text{old}}(s, a)) \leq 0$$

Similarly, for $\text{TD}^{\text{soft}} > 0$, we have :

$$0 \leq \beta \log \sum_a \exp(\frac{1}{\beta} Q^{\text{soft}}_{\text{new}}(s, a)) - \beta \log \sum_a \exp(\frac{1}{\beta} Q^{\text{soft}}_{\text{old}}(s, a)) \leq \pi_{\text{new}}(a_k|s_k)\text{TD}^{\text{soft}}$$

Thus, we have the upper bounds for $|\text{EVB}^{\text{soft}}|$,

$$\left|\text{EVB}^{\text{soft}}(e_k)\right| \leq \max\{\pi_{\text{old}}(a_k|s_k), \pi_{\text{new}}(a_k|s_k)\} * \left|\text{TD}^{\text{soft}}\right|.$$

For $|\text{PIV}^{\text{soft}}|$, we have,

$$\begin{aligned}
|\text{PIV}^{\text{soft}}(e_k)| &= |\sum_a \pi_{\text{new}}(a|s)\{q_{\text{new}}(s, a) - \beta \log(\pi_{\text{new}}(a|s))\} \\
&\qquad - \sum_a \pi_{\text{old}}(a|s)\{q_{\text{new}}(s, a) - \beta \log(\pi_{\text{old}}(a|s))\}| \\
&= \sum_a \pi_{\text{new}}(a|s)\{q_{\text{new}}(s, a) - \beta \log(\pi_{\text{new}}(a|s))\} \\
&\qquad - \sum_a \pi_{\text{old}}(a|s)\{q_{\text{old}}(s, a) - \beta \log(\pi_{\text{old}}(a|s))\} - \pi_{\text{old}}(a_i|s)\text{TD}^{\text{soft}} \\
&= \beta \log \sum \exp \frac{Q^{\text{soft}}_{\text{new}}(s, a)}{\beta} - \beta \log \sum \exp \frac{Q^{\text{soft}}_{\text{old}}(s, a)}{\beta} - \pi_{\text{old}}(a_i|s)\text{TD}^{\text{soft}},
\end{aligned}$$

where the second line is because the policy improvement value is always greater than or equal to 0 and by reordering.

For $\text{TD}^{\text{soft}} > 0$, we have:

$$0 \leq \beta \log \sum \exp \frac{Q^{\text{soft}}_{\text{new}}(s, a)}{\beta} - \beta \log \sum \exp \frac{Q^{\text{soft}}_{\text{old}}(s, a)}{\beta} - \pi_{\text{old}}(a_i|s)\text{TD}^{\text{soft}} \leq \pi_{\text{new}}(a_k|s_k)\text{TD}^{\text{soft}}$$

For $\text{TD}^{\text{soft}} \leq 0$, we have:

$$0 \leq \beta \log \sum \exp \frac{Q^{\text{soft}}_{\text{new}}(s, a)}{\beta} - \beta \log \sum \exp \frac{Q^{\text{soft}}_{\text{old}}(s, a)}{\beta} - \pi_{\text{old}}(a_i|s)\text{TD}^{\text{soft}} \leq \pi_{\text{old}}(a_k|s_k)\text{TD}^{\text{soft}}$$

Thus, we have the upper bounds for $\left|\text{PIV}^{\text{soft}}\right|$:

$$\left|\text{PIV}^{\text{soft}}\right| \leq \max\{\pi_{\text{old}}(a_k|s_k), \pi_{\text{new}}(a_k|s_k)\} * \left|\text{TD}^{\text{soft}}\right|$$

Also, for $|\text{EIV}^{\text{soft}}|$, we have:

$$\begin{aligned}
|\text{EIV}(e_k)| &= |\sum_a \pi_{\text{old}}(a|s)[Q^{\text{soft}}_{\pi_{\text{new}}}(s, a) - Q^{\text{soft}}_{\pi_{\text{old}}}(s, a)]| \\
&= \pi_{\text{old}}(a|s) * \left|\text{TD}^{\text{soft}}\right| \\
&\leq \max\{\pi_{\text{old}}(a_k|s_k), \pi_{\text{new}}(a_k|s_k)\} * \left|\text{TD}^{\text{soft}}\right|
\end{aligned}$$

□

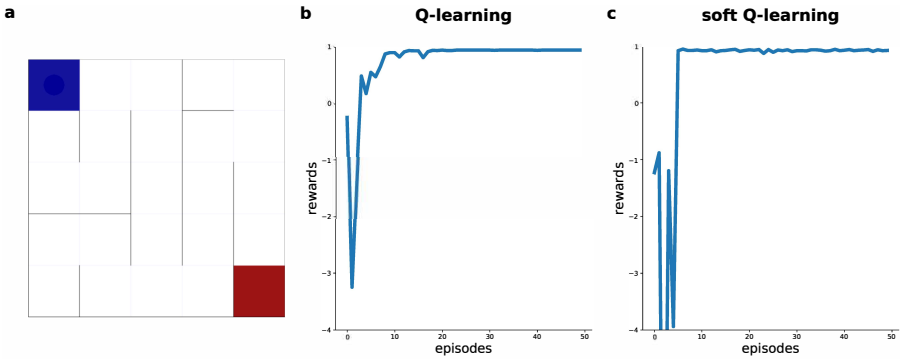

Figure 4: Maze environment and learning curves.

A.2 Experimental Details

A.2.1 Maze

For the maze experiments in section 5.1, we use a maze environment of a 5×5 square with walls, as depicted in Figure 4a. The agent needs to reach the goal zone in the bottom-right corner. At each time step, the agent can choose to move one square in any of the four directions (north, south, east, west). If the move is blocked by a wall or the border of the maze, the agent stays in place. Every time step, the agent gets a reward of -0.004 or 1 if it enters the goal zone and the episode ends. The discount factor is 0.99 throughout the experiments. For these experiments, we use a tabular setting for Q-learning and soft Q-learning according to section 2.1 and 4.1. For Q-learning, the behavior policy is ϵ-greedy, where ϵ decays exponentially from 1 to 0.001 during training. And we set learning step size $\alpha = 1$. For soft Q-learning, the temperature parameter β is set to 100. Total trial number is 50 for each algorithm. During training, both algorithms successfully solve the maze game, see Figure 4b-c for the learning curves.

A.2.2 CartPole

For CartPole, the goal is to keep the pole balanced by moving the cart forward and backward for 200 steps. We test our theoretical prediction on DQN and soft-DQN (DQN with soft-update). For DQN, we implement the model according to Mnih et al. (2015), where we replace the original Q-network with a two-layer MLP, with 256 Relu neurons for each layer. The ϵ in ϵ-greedy policy decays exponentially from 1 to 0.01 for the first $10,000$ steps, and remains 0.01 afterwards. For soft-DQN, all settings are the same with DQN, except for two modifications: for policy evaluation, the (soft) Q-network is updated according to the soft TD error; the policy follows maximum-entropy policy, calculated as the softmax of the soft Q values (see section 4.1). The temperature parameter β is set to 0.5. For both algorithms, the discount factor is 0.99, the learning rate is 0.005, experience buffer size is 1000, the batch size is 16 and total environment interaction is $50,000$.

A.2.3 Atari Games

For this set of experiments, we compare the performance of vanilla soft DQN and soft DQN with PER, where we use *surprise* and the derived upper bound as priorities (Schaul et al., 2016), respectively denoted as PER and VER (valuable experience replay). We select 9 Atari games for the experiments: Alien, BattleZone, Boxing, BeamRider, DemonAttack, MsPacman, Qbert, Seaquest and SpaceInvaders. The vanilla soft DQN is similar to that described in the above section, but the Q-network the same with Mnih et al. (2015). We implement PER on soft-DQN according to Schaul et al. (2016). For all algorithms, the temperature parameter β is 0.05, the discount factor is 0.99, the learning rate is $1\mathrm{e}{-4}$, experience buffer size is 1M, the batch size is 32, total environment interaction is $50,000$. For PER or VER, the parameters for importance sampling are $\alpha_{\text{IS}} = 0.4$ and $\beta_{\text{IS}} = 0.6$. For each game, the network is trained on a single GPU for 40M frames, or approximately 2 days.