HIGHWAY REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Traditional Dynamic Programming (DP) approaches suffer from slow backward credit-assignment (CA): only a one-step search is performed at each update. A popular solution for multi-step CA is to use multi-step Bellman operators. Unfortunately, in the control settings, existing methods typically suffer from the large variance of multi-step off-policy corrections or are biased, preventing convergence. To overcome these problems, we introduce a novel multi-step Bellman optimality equation with adaptive lookahead steps. We first derive a new multi-step Value Iteration (VI) method that converges to the optimal Value Function (VF) with an exponential contraction rate but linear computational complexity. Given some trial, our so-called *Highway RL* performs rapid CA, by picking a policy and a possible lookahead (up to the trial end) that maximize the near-term reward during lookahead plus a DP-based estimate of the cumulative reward for the remaining part of the trial. Highway RL does not require off-policy corrections. Under mild assumptions, it achieves better convergence rates than the traditional one-step Bellman Optimality Operator. We then derive Highway Q-Learning, a convergent multi-step off-policy variant of Q-learning. We show that our Highway algorithms significantly outperform DP approaches on toy tasks. Finally, we propose a deep function approximation variant called Highway DQN. We evaluate it on visual MinAtar Games, outperforming similar multi-step methods.

1 INTRODUCTION

Recent advances in multi-step reinforcement learning (RL) have achieved remarkable empirical success (Horgan et al., 2018; Barth-Maron et al., 2018). However, a major challenge of multi-step RL is to balance the trade-off between traditional "safe" one-time-step-per-trial credit assignment (CA) relying on knowledge stored in a learned Value Function (VF), and large CA jumps across many time steps. A traditional way of addressing this issue is to impose a fixed prior distribution over the possible numbers of CA steps, e.g., TD(λ) (Sutton & Barto, 2018), GAE(λ) (Schulman et al., 2016). This typically ignores the current state-specific quality of the current VF, which dynamically improves during learning. Besides, the prior distribution usually has to be tuned case by case.

Multi-step RL should also work for off-policy learning, that is, learning from data obtained by other behavioral policies. Most previous research on this has focused on Policy Iteration(PI)-based approach (Sutton & Barto, 2018), with the goal of evaluating the VF of a target policy (Precup, 2000; Harutyunyan et al., 2016; Munos et al., 2016; Sutton & Barto, 2018; Schulman et al., 2016). PI-based methods need to correct the discrepancy between target policy and behavior policy (Sutton & Barto, 2018). A classic method is importance sampling(IS)-based correction, which is proven to be unbiased theoretically. However, classic IS-based methods usually suffer from certain undesirable side effects of high variance due to product of importance sampling (IS) ratios. Recently, several variance reduction methods have been proposed and shown to be effective in practice, such as Q(λ) (Harutyunyan et al., 2016), Retrace (λ) (Munos et al., 2016) and C-trace (Rowland et al., 2020), and so on (Espeholt et al., 2018; Horgan et al., 2018; Asis et al., 2017). In contrast to PI, Value Iteration (VI) methods aim at approximating the optimal VF by propagating the values of the most promising actions backwards one step at a time (Sutton & Barto, 2018; Szepesvári, 2010). Such methods can safely use data from any behavioral policy without any correction. However, step-by-step value propagation makes them somewhat ill-suited for general multi-step CA.

We propose a novel multi-step Bellman Optimality Operator with adaptive lookahead step that reflects the latent structure of multi-step CA within the optimal VF framework. Our Highway operator

adaptively selects a policy and a lookahead that maximize the near-term reward during lookahead plus a DP-based estimate of the cumulative reward for the remaining part of the trial. It has the following desirable properties: 1) It converges to the optimal VF V^* with minor requirements; 2) It safely uses arbitrary off-policy data, without requiring off-policy corrections; 3) Under mild assumptions it achieves better convergence rates than traditional Bellman Optimality Operators; 4) It effectively assigns credit over multiple time-steps.

Based on this new operator, we derive new multi-step versions of Value Iteration, Q-Learning and DQN. Experiments on toy tasks and on visual MinAtar Games (Young & Tian, 2019)illustrate that our *Highway RL* algorithms outperform DP and multi-step methods.

2 PRELIMINARIES

A *Markov Decision Processes (MDP)* (Puterman, 2014) is described by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \gamma, \mathcal{T}, \mu_0, \mathcal{R})$, where $\mathcal{S}$ is the state space; $\mathcal{A}$ is the action space; $\gamma \in [0, 1)$ is the discount factor. We will assume MDPs with countable $\mathcal{S}$ (discrete topology) and finite $\mathcal{A}$. $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition probability function; μ_0 denotes the initial state distribution; $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathbb{R})$ denotes reward probability function. We will use the following symbols to denote related conditional probabilities: $\mathcal{T}(s'|s, a)$, $\mathcal{R}(\cdot|s, a)$, $s, s' \in \mathcal{S}$, $a \in \mathcal{A}$. We will also use $r(s, a) \triangleq \mathbb{E}_{R\sim\mathcal{R}(\cdot|s,a)}[R]$ for convenience. We need the space of value functions to be complete (assumption of Banach fixed point theorem). Therefore we will assume bounded rewards, which with discounting gives us bounded value functions. We will denote $l_\infty(\mathcal{X})$ the space of bounded sequences with supremum norm $\|\cdot\|_\infty$ with support $\mathcal{X}$ assuming $\mathcal{X}$ is countable and has discrete topology. Completeness of our value spaces then follows from completeness of $l_\infty(\mathbb{N})$ [1].

The goal is to find a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ that yield maximal return, defined as the accumulated discounted reward from time step t, i.e., $G_t = \sum_{n=0}^{\infty} \gamma^n r(s_{t+n}, a_{t+n})$. The state-value function (VF) of a policy π is defined as the expected return of being in state s and following policy π, $V^\pi(s) \triangleq \mathbb{E}[G_t|s_t = s; \pi]$. Let Π denote the space of π. The *optimal VF* is $V^*(s) = \max_{\pi\in\Pi} V^\pi(s)$. It is also convenient to define the action-VF, $Q^\pi(s, a) \triangleq \mathbb{E}[G_t|s_t = s, a_t = a; \pi]$ and the optimal action-VF is denoted as $Q^* = \max_{\pi\in\Pi} Q^\pi(s)$. The Bellman Expectation/Optimality Equation and the corresponding operators are as follows:

$$\mathcal{B}^\pi V^\pi = V^\pi, \text{ where } (\mathcal{B}^\pi V)(s) \triangleq \mathbb{E}_{a\sim\pi(\cdot|s), s'\sim\mathcal{T}(\cdot|s,a)} [r(s, a) + \gamma V(s')] \tag{1}$$

$$\mathcal{B}V^* = V^*, \text{ where } (\mathcal{B}V)(s) \triangleq \max_a \left[r(s, a) + \gamma\mathbb{E}_{s'\sim\mathcal{T}(\cdot|s,a)}[V(s')]\right]. \tag{2}$$

3 HIGHWAY REINFORCEMENT LEARNING

Dynamic Programming (DP) looks ahead for one step to identify a promising value using only short-term information. How can we quickly exploit long-term information through larger lookaheads? Our idea is to exploit the information conveyed by policies. We connect current and future states through multiple "highways" induced by policies, allowing for the unimpeded flow of credit across various lookaheads, and then focus on the most promising highway. Formally, we propose the novel Highway Bellman Optimality Operator $\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}$ (Highway Operator in short), to update the VF in the following way:

$$\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} V(s_0) \triangleq \max_{\pi\in\widehat{\Pi}} \max_{n\in\mathcal{N}} \mathbb{E}_{\tau_{s_0}^n\sim\pi} \left[\sum_{t=0}^{n-1} \gamma^t r(s_t, a_t) + \gamma^n \max_{a'_n} \left[r(s_n, a'_n) + \gamma\mathbb{E}_{s'_{n+1}} \left[V\left(s'_{n+1}\right)\right]\right]\right], \tag{3}$$

where $\widehat{\Pi} = \{\pi_1, \cdots, \pi_M\}$ is a *set of behavioral policies*, which are used to collect data; $\mathcal{N}$ is the *set of lookahead steps* (also named bootstrapping steps in RL literature) which includes 0, i.e., $0 \in \mathcal{N}$; let $N = \max_{n\in\mathcal{N}}$ denote the *maximal lookahead step*; $\tau_{s_0}^n = (s_0, a_0, s_1, a_1, s_2, a_2, \cdots, s_n)$, and $\tau_{s_0}^n \sim \pi$ is the trajectory starting from s_0 by executing policy π for N steps. Fig. 1 (Left) illustrates

[1] The space of all bounded sequences with supremum norm which is known to be complete metric space.

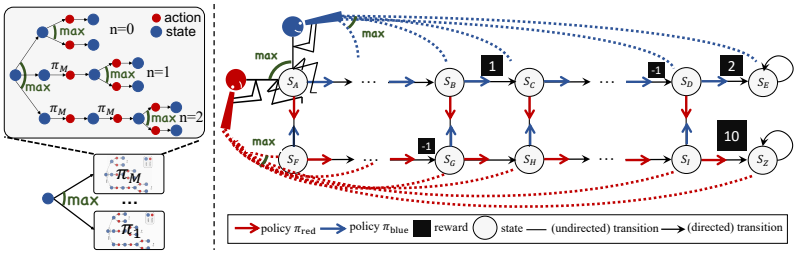


Figure 1: (Left) Backup diagram of Highway Operator $\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}$ with $\mathcal{N} = \{0, 1, 2\}$. (Right) An illustrative example of "highway" under a simple N-horizon MDP, where N is the horizon between the start state s_A and the end state s_Z. With the highway (dashed lines) induced by rolling-out behavioral policies, the start state s_A can directly access various deeper horizon information. By taking the maximization over these information, the high credit information can be directly assigned to the previous states through the highway.

the backup diagram of this operator. Our operator can be rewritten using the Bellman Operators:

$$\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} V \triangleq \max_{\pi \in \widehat{\Pi}} \max_{n \in \mathcal{N}} \left(\mathcal{B}^{\pi}\right)^{n} \mathcal{B} V. \tag{4}$$

As implied by eq. (3), given some trial, we pick a policy and a possible lookahead (up to the trial end if N is sufficiently large) that maximize the cumulative reward during the lookahead interval plus a dynamic programming-based estimate of the future cumulative reward during the remaining trial. This provides a highway to quickly transport credit information from the future into the past. That's why our operator generally converges faster than the classical Bellman Optimality Operator. See the detailed theoretical analysis in Section 5. Fig. 1 (Right) illustrates this highway through an example: an N-horizon MDP problem and two policies. A highway connects the start state s_A to information derived from deeper lookaheads. In this example, our operator converges to the optimal VF within 2 iterations, while $\mathcal{B}$ needs N.

The following result shows that our Highway operator is a contraction on the complete metric space l_∞ and admits V^* as unique fixed point. Moreover, the sequence of value functions produced by the application of the operator have convergence rate γ:

Theorem 1 *(Highway Bellman Optimality Equation (Highway Equation)) Let us have an MDP $\mathcal{M}$. For any $\widehat{\Pi}$ and $\mathcal{N}$ (s.t. $0 \in \mathcal{N}$), we have*

1) $\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}$ is a contraction on complete metric space $l_\infty(\mathcal{S})$, i.e. for any $V, V' \in l_\infty(\mathcal{S})$, we have[2] $\|\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} V - \mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} V'\| \leq \|\mathcal{B}V - \mathcal{B}V'\| \leq \gamma \|V - V'\|$;

2) V^* is the only fixed point $\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}$. That is,

$$\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} V^* = V^*, \tag{5}$$

and for all $V \in l_\infty(S)$ holds $V = \mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} V$ if and only if $V = V^*$.

3) For any $V_0 \in l_\infty(\mathcal{S})$ and any sequence of set of behavioral policies $(\widehat{\Pi}_1, \cdots, \widehat{\Pi}_k)$, the sequence $(\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}_k} \circ \mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}_{k-1}} \circ \ldots \circ \mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}_1})[V_0], k \in \mathbb{N}$ converge R-linearly to V^* with convergence rate γ.

All the proofs are provided in Appendix A. Table 1 summarizes the comparison to classical Bellman Operators and some advanced operators. More details on the comparison are in Section 5.

4 Algorithms

Our Highway Operator provides a powerful extension of the theory of Bellman Optimality, which can be potentially applied to various RL control methods. Here we illustrate applications of our Highway theory for multi-step Value Iteration, multi-step Q-learning and multi-step DQN.

[2] We denote by $\|\cdot\|$ for supremum norm throughout the paper.

Operator	Fixed Point	Guaranteed convergence	Contraction Rate	NOT suffering variance explosion of IS ratios
One-Step Bellman Optimality Operator $\mathcal{B}$	V^*	/	γ	✓
Multi-Step Bellman Optimality Operator $\mathcal{B}_N^{\widehat{\Pi}}$ $(N \geq 2)$	V^*	† For $\widehat{\Pi}$ satisfies $\forall \pi' \in \widehat{\Pi}, \pi' = \pi^*$	γ^N	✓
Multi-Step IS-based Bellman Expectation Operator $\mathcal{B}_N^{\pi,\widehat{\Pi}}$	V^π	For any $\widehat{\Pi}$	γ^N	×
Q(λ) Operator (Harutyunyan et al., 2016)	V^π	For $\widehat{\Pi}$ satisfies $\forall \pi' \in \widehat{\Pi}$, π' is close to π	γ	✓
Retrace(λ) Operator (Munos et al., 2016)	V^π	For any $\widehat{\Pi}$	γ	✓
Highway Operator $\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}$ (our)	V^*	For any $\widehat{\Pi}, \mathcal{N}$ (s.t. $0 \in \mathcal{N}$)	γ (γ^2, γ^N under some conditions)	✓

Table 1: Properties of the off-policy operators. $\widehat{\Pi}$ denote the set of behavioral policies; π denote the target policy for Policy Iteration-based methods. †: Please refer to Appendix A.2 for details.

4.1 Dynamic Programming

Highway Value Iteration. From the new operator we can naturally derive a new Value Iteration algorithm, (Algorithm 1 in APPENDIX). Specifically, for a discrete MDP, the update using the Highway Operator $\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}$ (eq. 3) can be written as

$$\mathbf{v}_{k+1} = \max_{\pi \in \widehat{\Pi}} \max_{n \in \mathcal{N}} \overbrace{\left[\underbrace{\sum_{i=0}^{n} \left[(\gamma \mathbf{T}^\pi)^{i-1} \right] \mathbf{r}^\pi}_{\mathbf{PART1}} + \underbrace{(\gamma \mathbf{T}^\pi)^n}_{\mathbf{PART2}} \left[\max_a \left[\mathbf{r}^a + \gamma \mathbf{T}^a \mathbf{v}_k \right] \right] \right]}^{\mathbf{PART3}}, \tag{6}$$

where $\mathbf{v}_k$ is a $|\mathcal{S}| \times 1$ column vector of VF; $\mathbf{r}^a$ and $\mathbf{r}^\pi$ are $|\mathcal{S}| \times 1$ column vectors of rewards for action a and policy π respectively, where $[\mathbf{r}^a]_s = r(s,a)$, $[\mathbf{r}^\pi]_s = \sum_a \pi(a|s) r(s,a)$. $\mathbf{T}^a$ and $\mathbf{T}^\pi$ are $|\mathcal{S}| \times |\mathcal{S}|$ matrices of transition probabilities for action a and policy π respectively, where $[\mathbf{T}^a]_{s,s'} = \mathcal{T}(s'|s,a)$, $[\mathbf{T}^\pi]_{s,s'} = \sum_a \pi(a|s) \mathcal{T}(s'|s,a)$. The computational complexity of each iteration is $\mathcal{O}\left(\left(|\mathcal{A}| + \left| \widehat{\Pi} \right| |\mathcal{N}| \right) |\mathcal{S}|^2 \right)$. Two strategies can be adopted to accelerate the update process. First, the matrix in **PART 1** and **PART 2** in eq. (6) can be computed and reused for each π and n, as they are fixed during the iteration process. Second, **PART 3** can be computed in parallel for each policy $\pi \in \widehat{\Pi}$ and $n \in \mathcal{N}$.

4.2 Off-Policy Learning in model-free RL

In model-free RL, it is convenient to use Q instead of V. The corresponding operator[3] is defined as

$$\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} Q(s_0, a_0) \triangleq \max_{\pi \in \widehat{\Pi}} \max_{n \in \mathcal{N}} \mathbb{E}_{\tau_{s_0,a_0}^{n+1} \sim \pi} \underbrace{\left[\sum_{t=0}^{n} \gamma^t r_t + \gamma^{n+1} \max_{a'_{n+1}} Q(s_{n+1}, a'_{n+1}) \right]}_{G_Q^{n+1}(\tau_{s_0,a_0}^{n+1})}, \tag{7}$$

where $G_Q^{n+1}(\tau_{s_0,a_0}^{n+1})$ is the $n+1$-step return; $\tau_{s_0,a_0}^{n+1} \triangleq (s_0, a_0, r_0, s_1, a_1, r_1, \cdots, s_{n+1})$. This operator can also be represented by the Bellman Operators, i.e., $\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} Q \triangleq \max_\pi \max_n (\mathcal{B}^\pi)^n \mathcal{B} Q$; and it converges to the optimal action VF Q^* with any set of behavioral policies $\widehat{\Pi}$, i.e., $\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} Q^* = Q^*$ (similar to Theorem 1, see Theorem 6 in the APPENDIX). This means that we can utilize any off-policy data collected by arbitrary policy, without additional corrections. We propose two methods for the tabular VF and VF approximation, named Highway Q-Learning and Highway DQN respectively.

Highway Q-Learning. Let $\mathcal{D}_{s_0,a_0}^{(m)} = \{\tau_{s_0,a_0}^{n+1} | \tau_{s_0,a_0}^{n+1} \sim \pi_m\}$ denote the trajectory data collected by the policy π_m. The k-th VF Q_k is updated in the following way:

[3]We use the same notation $\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}$ to denote the operator w.r.t. the the VF V and the action VF Q when there is no ambiguity. Similarly, for the Bellman operators we will reuse the same symbols.

$$Q_{k+1}\left(s_0, a_0\right) = \max_{m \in \mathsf{M}_{s_0,a_0}} \max_{n \in \mathcal{N}} \widehat{\mathbb{E}}^{\mathcal{D}^{(m)}_{s_0,a_0}} [G^{n+1}_{Q_k}(\tau^{n+1}_{s_0,a_0})], \tag{8}$$

where $\mathsf{M}_{s_0,a_0} = \left\{ m \middle| |\mathcal{D}^{(m)}_{s_0,a_0}| \neq 0 \right\}$ are the indexes of the dataset that are not empty under (s_0, a_0); $\widehat{\mathbb{E}}^{\mathcal{D}^{(m)}_{s_0,a_0}}[\cdot] = \frac{1}{\left|\mathcal{D}^{(m)}_{s_0,a_0}\right|} \sum_{\tau^{n+1}_{s_0,a_0} \in \mathcal{D}^{(m)}_{s_0,a_0}} [\cdot]$ is the empirical averaged value. Note that all we need to do is saving the trajectory data into the corresponding dataset $\mathcal{D}^{(m)}_{s_0,a_0}$, and then search over it, without having to know the form of π_m or save the π_m into the set of behavioral policies $\widehat{\Pi}$. The algorithm, *Highway Q learning*, is presented in Appendix B Algorithm 3.

Highway DQN. For large-scale space or continuous MDPs, the VF is usually approximated. Due to the estimation noise involved in function approximation (Van Hasselt et al., 2016), our method may lead to overestimation by the two maximization operations (over policies and lookahead steps). However, as we show below, this issue can be easily solved through a minor modification.

We propose a new variant of eq. (7), named Softmax Highway Operator, as follows

$$\widetilde{\mathcal{G}}^{\widehat{\Pi}}_{\mathcal{N}} Q\left(s_0, a_0\right) \triangleq \underset{\pi \in \widehat{\Pi}}{smax^{\alpha}} \underset{n' \in \mathcal{N}}{smax^{\alpha}} \max_{n \in \{0, n'\}} \mathbb{E}_{\tau^{n+1}_{s_0,a_0} \sim \pi} \left[G^{n+1}_{Q}(\tau^{n+1}_{s_0,a_0}) \right], \tag{9}$$

where the softmax operator $smax^{\alpha}$ with the temperature parameter $\alpha \in [0, +\infty)$ is defined as

$$\underset{x \in \mathcal{X}}{smax^{\alpha}} f\left(x\right) \triangleq \sum_{x \in \mathcal{X}} \frac{\exp\left(\alpha f\left(x\right)\right)}{\sum_{x' \in \mathcal{X}} \exp\left(\alpha f\left(x'\right)\right)} f\left(x\right), \tag{10}$$

$smax^{\alpha}$ reduces to $\max$ when $\alpha \to \infty$. We have the following theorem.

Theorem 2 *For any α, any $\widehat{\Pi}$, and any $\mathcal{N}$, $\widetilde{\mathcal{G}}^{\widehat{\Pi}}_{\mathcal{N}}$ is a contraction and $\widetilde{\mathcal{G}}^{\widehat{\Pi}}_{\mathcal{N}} Q^* = Q^*$ holds.*

The operator in eq. (9) is derived in the following way. First, the Highway Q Operator $\mathcal{G}^{\widehat{\Pi}}_{\mathcal{N}}$ in eq. (7) can be rewritten as an equivalent form, $\mathcal{G}^{\widehat{\Pi}}_{\mathcal{N}} Q \triangleq \max_{\pi} \max_{n'} \max_{n \in \{0, n'\}} (\mathcal{B}^{\pi})^n \mathcal{B}$. Then, we replace the first two max operators $\max_{\pi} \max_{n'}$ with the softmax operators $smax^{\alpha}_{\pi}\, smax^{\alpha}_{n'}$.

The above modification of Softmax Highway Operator $\widetilde{\mathcal{G}}^{\widehat{\Pi}}_{\mathcal{N}}$ is necessary to a) remain unbiased w.r.t. Q^* (as shown in Theorem 2)[4] ; b) alleviate the overestimation issue and improve exploration with the softmax operation, which has been shown effective in recent RL literature (Fox et al., 2015; Haarnoja et al., 2017; Schulman et al., 2017; Song et al., 2019).

Based on the above theoretically justified operator, we propose the following objective function for updating the action VF $Q_{\theta}(s, a)$ parametrized by the parameter θ:

$$L\left(\theta\right) = \sum_{(s_0,a_0) \in \widehat{\mathcal{D}}} \left[Q_{\theta}\left(s_0, a_0\right) - \underset{m \in \mathsf{M}_{s_0,a_0}}{smax^{\alpha}} \underset{n' \in \mathcal{N}}{smax^{\alpha}} \max_{n \in \{0, n'\}} \widehat{\mathbb{E}}^{\mathcal{D}^{(m)}_{s_0,a_0}} \left[G^{n+1}_{Q_{\theta'}}(\tau^{n+1}_{s_0,a_0}) \right] \right]^2, \tag{11}$$

where $Q_{\theta'}$ is the target network parametrized by θ', occasionally copied from θ; $\widehat{\mathcal{D}} = \{(s_0, a_0)\})$ is the sampled batch data of state-action (s_0, a_0) pairs. The computational complexity of the method implied by the equation above is close to the one of existing multi-step methods (Schulman et al., 2016; Precup, 2000; Harutyunyan et al., 2016; Munos et al., 2016), which also need to compute n-step returns G^n_Q for n steps. The resulting algorithm is called *Highway DQN*, presented in Appendix B Algorithm 4.

In summary, our Highway Q learning and Highway DQN can recycle any trajectory data collected by some arbitrary policy, and utilize the multiple-step trajectory data that do not require Importance Sampling-based corrections (as stated in Theorem 1 and 2).

[4]Note that another variant $smax^{\alpha}_{\pi}\, smax^{\alpha}_{n} (\mathcal{B}^{\pi})^n \mathcal{B}$ is generally biased w.r.t. Q^*.

5 THEORETICAL ANALYSIS

First we compare our Highway Operator $\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}$ to the Bellman Optimality Operator $\mathcal{B}$[5].

Theorem 3 *(Comparison to Bellman Optimality Operator $\mathcal{B}$)*
1) $\left\|\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} V - V^*\right\| \leq \|\mathcal{B}V - V^*\|$*;*
2) Assume that $V_0 \leq V^*$*: for any* s *we have* $\left|\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} V_0(s) - V^*(s)\right| \leq |\mathcal{B}V_0(s) - V^*(s)|$*; the inequality holds as long as there exists* $\pi' \in \widehat{\Pi}$ *such that* $\arg\max_{n\in\mathcal{N}} \left(\mathcal{B}^{\pi'}\right)^n \mathcal{B}V_0(s) > 0$.

The first point of the theorem implies that our operator converges at the same rate as the Bellman Optimality Operator in the worst case. The second point shows a state-wise convergence comparison under the case of $V_0 \leq V^*$. Our operator generally converges faster than the Bellman Optimality Operator as long as one behavioral policy finds a better path by looking forward for n steps ($n > 0$). Note that the condition $V_0 \leq V^*$ can be easily satisfied by setting $V_0 = \min_{s',a'} r(s', a')$. Moreover, as long as $V_0 \leq V^*$ we have $(\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}})^{\circ k} V_0 \leq V^*$ for any k (see Lemma 2 in Appendix).

We then show that our operator provides an optimal solution for the evaluation-improvement trade-off to Generalized Policy Iteration (GPI) (see APPENDIX Algorithm 2) (Sutton & Barto, 2018). GPI interacts between policy improvement and policy evaluation by applying $\mathcal{B}^\pi$ for N times. Our operator selects the best N to approximate the optimal VF V^* under $V \leq V^*$.

Theorem 4 *(Comparison to Multi-Step Bellman Expectation Operator) Assume that the VF* $V_k \leq V^*$*. Let* $V_{k+1}^{\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}_k}}$ *and* $V_{k+1}^{\mathcal{B}_N^{\pi_k}}$ *denote the* $k+1$*-th VF of Highway Value Iteration with hyperparameter* $\mathcal{N}$ *and Generalized Policy Iteration with hyperparameter* N*. We have*

$$\left\|V_{k+1}^{\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}_k}} - V^*\right\| \leq \min_{N\in\mathcal{N}} \left\|V_{k+1}^{\mathcal{B}_{N+1}^{\pi_k}} - V^*\right\| \tag{12}$$

We now show that by assuming that some of the behavioral policies act optimally for a few time steps, we can achieve higher convergence rates.

Assumption A$(\widehat{\Pi}, n)$ *Given set of behavioral policies* $\widehat{\Pi}$*, a lookahead step* n*. Let* $\mathcal{S}_{s,n}^{\pi'}$ *denote the set of all possible visited states by executing policy* π' *for* n *steps, starting from state* s*. We have: for each* $s \in \mathcal{S}$ *there exists at least one policy* $\pi' \in \widehat{\Pi}$ *such that* $\pi'(s') = \pi^*(s')$ *for any* $s' \in \mathcal{S}_{s,n}^{\pi'}$ *and* π^* *an optimal policy. Note the quantification order:* $(\forall s \in \mathcal{S}, \exists \pi' \in \widehat{\Pi}, \forall s' \in \mathcal{S}_{s,n}^{\pi'}) : \pi'(s') = \pi^*(s')$.

Theorem 5 *(Better Contraction Rate) Assume* $N-1 \in \mathcal{N}$*,* $\widehat{\Pi}$ *satisfies Assumption A*$(\widehat{\Pi}, N-1)$*, and* $V_0 \in l_\infty(\mathcal{S})$ *and* $V_0 \leq V^*$*. the convergence rate of* $\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}$ *is* γ^N*, i.e.,* $\|\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} V - V^*\| \leq \gamma^N \|V - V^*\|$*;*

This theorem implies that when the set of behavioral policies $\widehat{\Pi}$ satisfies Assumption A$(\widehat{\Pi}, N-1)$, our operator $\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}$ has a convergence rate of γ^n. Note that Assumption A$(\widehat{\Pi}, N-1)$ does not require the $\widehat{\Pi}$ includes the optimal policy π^*. Instead, it only requires that, for each state, there exists one policy that behaves well within a period ($N-1$ consecutive steps), and the well-behaved policy could be different starting from each state. This assumption is reasonable for two reasons: 1) Over the policy improvement process, more asymptotically optimal policies are added to the set of behavioral policies (for more details please refer to APPENDIX Lemma 1 and corresponding discussion). 2) In tasks where there exist sub-tasks (or sub-goals), the agent can easily obtain local-optimal policies for solving the sub-tasks. Specifically, Assumption A$(\widehat{\Pi}, 1)$ can be satisfied by constructing a $\widehat{\Pi}' = \{\pi_i | \pi_i(s) = a_i, i = 1, \cdots, |\mathcal{A}|\}$ yielding a convergence rate of γ^2.

[5]Note that, unless otherwise stated, the results hold for any set of behavioral policies $\widehat{\Pi}$ and set of lookahead steps $\mathcal{N}$ (s.t. $0 \in \mathcal{N}$). Note that in the process of Highway Value Iteration (see APPENDIX Algorithm 1), the set of behavioral policies $\widehat{\Pi}_k$ could change over iterations as new policies are added to the set of behavioral policies. Theorem 1 point *3)* in fact gives a general result for such a case with a dynamically changing set of behavioral policies.

Our operator $\max_{\pi \in \widehat{\Pi}} \max_{n \in \mathcal{N}} (\mathcal{B}^{\pi})^n \mathcal{B}$ is different from applying the Bellman Optimality Operator $\mathcal{B}$ for N times, i.e., $(\mathcal{B})^{\circ N}$. We searches over the $\widehat{\Pi}$ and the $\mathcal{N}$, therefore its computational complexity is $\mathcal{O}\left(\left(|\mathcal{A}| + \left|\widehat{\Pi}\right| |\mathcal{N}|\right) |\mathcal{S}|^2\right)$. On the other hand, $(\mathcal{B})^{\circ N}$ searches over the action space for N times, therefore its computational complexity is $\mathcal{O}\left(|\mathcal{A}|\, N |\mathcal{S}|^2\right)$. These two operators work typically different in model-free case: our operator is able to use full trajectory data (see Section 4.2), while $(\mathcal{B})^{\circ N}$ implies much greater computational cost.

6 Related Work

Most multi-step RL methods are policy iteration-based methods with interacting policy improvements and target policy evaluations. This naturally allows for multi-step CA. Related algorithms include multi-step SARSA (Sutton & Barto, 2018), Tree Backup (Precup, 2000), Q(σ) (Asis et al., 2017), and Monte Carlo methods (which can be regarded as ∞-step SARSA). Such N-step CA can be combined with weights exponentially decaying in N, which is known as λ-returns (Sutton & Barto, 2018; Schulman et al., 2016; White & White, 2016). Sahil et al. (2017) proposed a generalization of λ-returns called weighted-returns, which assigns a weight w_n to each n-step return. Roughly, the lookahead step n, the decay factor λ, and the weights w represent the prior knowledge or bias regarding appropriate CA intervals, usually tuned in a case-by-case manner. Our method instead adaptively adjusts the CA lookaheads in line with the quality of the data and the learned VF. Another similar work related to CA is RUDDER (Arjona-Medina et al., 2019), which trains an LSTM to re-assign credits to previous actions. Our methods derive a simple principle for transporting the credit without additional cost and can be flexibly extended to any value-based algorithms.

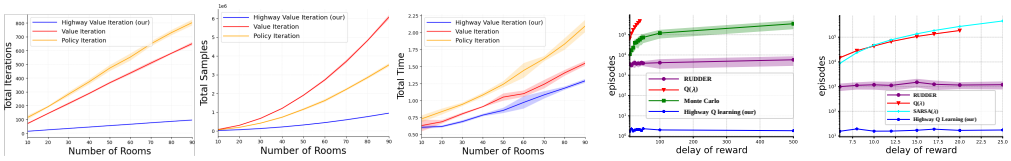
Policy evaluation-based methods usually require additional corrections for off-policy data collected from other policies. A classic is the importance sampling (IS) correction. Since the products of IS ratios may cause high variance (Cortes et al., 2010; Metelli et al., 2018), several variance reduction techniques have been proposed. Based on the Q(λ) method (Harutyunyan et al., 2016), Munos et al. (2016)'s Retrace(λ) reduces the variance by clipping the IS ratios. Retrace(λ) has achieved great success in practice. Other work ignores IS: TB(λ) (Precup, 2000) corrects the off-policy data by computing expectations w.r.t. data and estimated VF, using the probabilities of the target policy. Harutyunyan et al. (2016)'s Q(λ) corrects the off-policy data by adding an off-policy correction term. Our method provides an alternative IS-free tool for off-policy learning. Similar to Retrace(λ), it can safely use arbitrary off-policy data. Moreover, it is very efficient, offering a faster contraction rate under mild conditions. More research is needed, however, to understand how the variance of our method compares to that of advanced IS-based variance reduction methods like Retrace(λ).

There is also previous work on the VI idea of taking the most promising value among various candidates. Efroni et al. (2018) propose to search in action space over multiple steps. However, this is computationally very expensive, and the authors consider it as a surrogate MDP problem (Tomar et al., 2020). In contrast, our methods search over rollout data from various behavioral policies. Barreto et al. (2020) also propose an operator searching over the value functions of multiple candidate policies, i.e., $\max_{\pi} Q^{\pi}(s, a)$. Their method can be regarded as a special case of our method where $\mathcal{N} = \{\infty\}$, but it is not shown to converge to Q^* with an improved convergence rate of γ^n.

Our method can also be viewed as combining the best of both worlds: (1) direct policy search based on policy gradients (Williams, 1992) or evolutionary computation (Rechenberg, 1971) (where lookahead equals trial length——no runtime-based identification of useful subprograms and subgoals), and (2) dynamic programming-based (Bellman, 1957) identification of useful subgoals during runtime. This naturally and safely combines best subgoals/sub-policies (Schmidhuber, 1991) derived from data collected by arbitrary previously encountered policies. It can be viewed as a soft hierarchical chunking method (Schmidhuber, 1992) for rapid CA, with a sound Bellman-based foundation.

7 Experiments

We designed our experiments to compare our Highway algorithms to the classical methods under different cases and investigate their properties. Please refer to Appendix C for details of tasks, implementation and additional results.



(a) Multi-Room Environment (b) Choice (c) Trace Back

Figure 2: (a) shows results of model-based algorithms in Multi-Room environments. The x-axis is the number of rooms. The y-axes for the three figures are total iteration, total samples, and total running time required by the algorithm, respectively. (b) and (c) on Choice and Trace Back environments: number of episodes required to solve the task, as a function of the delay of reward. Average over 100 seeds, 1 standard deviation.

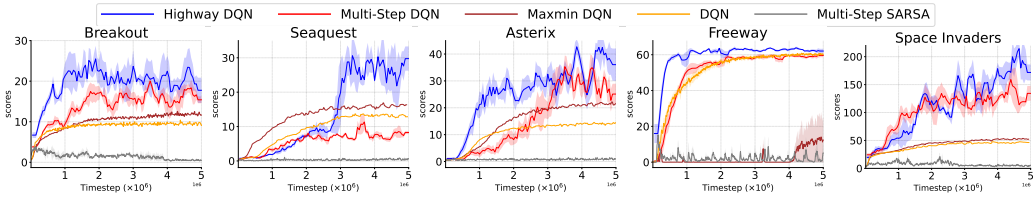
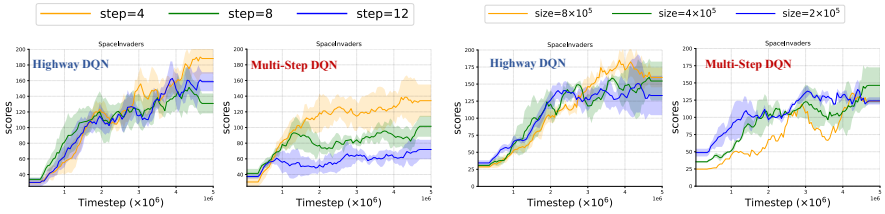


Figure 3: Episode rewards during the training on MinAtar Games (Young & Tian, 2019). Average over 5 seeds, 1 standard deviation.



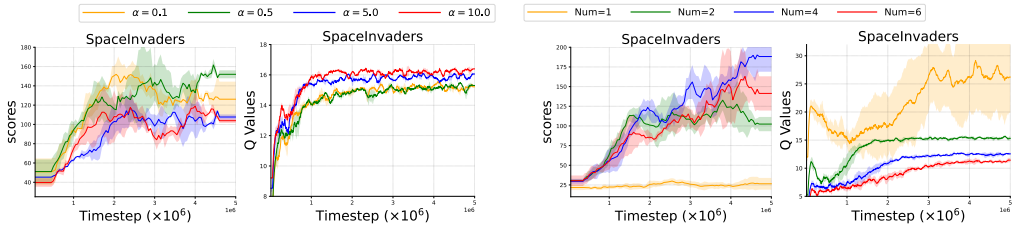
(a) Varying lookahead steps (b) Varying replay buffer sizes

Figure 4: The performance of Highway DQN and Multi-Step DQN using (a) varying lookahead steps and (b) varying replay buffer sizes. Average over 5 seeds, 1 standard deviation.

Model-based Toy Tasks. We first evaluate on a model-based task: Multi-Room environments with different number of rooms. The agent needs to go through many rooms and reach the goal to get a reward. We compare our Highway Value Iteration to classical Value Iteration (VI) and Policy Iteration (PI). The algorithms are evaluated until convergence. As shown in the three plots of Fig. 2 (a), our Highway Value Iteration outperforms VI and PI in terms of number of iterations required, total number of samples (total number of queries to the MDP model), and computation time.

Model-free Toy Tasks. To evaluate credit assignment efficiency, we evaluate the algorithms on two toy tasks involving delayed rewards (Arjona-Medina et al., 2019), in which a reward is only provided at the end of each trial. We compare our method to classical eligibility trace methods including Q(λ), SARSA(λ), Monte Carlo methods; and also the advanced credit assignment method RUDDER (Arjona-Medina et al., 2019). For our Highway Q-Learning, we set the maximal lookahead step $N = T - t$, the length from the current time step t until the end of the trial T.

As shown in Fig. 2 (b) and (c), our method significantly outperforms all competitors on both tasks. For example, on Trace Back, our method requires only 20 episodes to solve the task, while the second best algorithm RUDDER requires more than 1000. Notably, the costs of Highway Q learning do not observably increase with the reward delays. In contrast, other methods such as Q(λ) require exponentially increasing numbers of trials. Our Highway Q-Learning can quickly assign credit over multiple time steps.



(a) Performance (left) and Q values (right) using varying softmax temperatures.

(b) Performance (left) and Q values (right) using varying number of target network

Figure 5: (a) Performance (left) and Q values (right) of Highway DQN using varying softmax temperatures α. (b) Corresponding results using varying number of target networks. Average over 5 seeds, 1 standard deviation.

MinAtar Games. We evaluate our algorithms on benchmark tasks from MinAtar (Young & Tian, 2019). We implement our method on top of *Maxmin DQN* (Lan et al., 2020), which uses the smallest Q-value of several target networks as the estimate, i.e., $Q_{\theta'}(s,a) = \min_{1\leq u\leq U} Q_{\theta'_u}(s,a)$, where θ'_u is the parameter of u-th target network. We also compare to several advanced multi-step methods, including *Multi-Step DQN* (Horgan et al., 2018; Barth-Maron et al., 2018), and *Multi-Step SARSA* respectively. All proposed methods adopt the same implementations to ensure that we measure differences between algorithms, not implementations.

Fig. 3 shows the performance of the algorithms. Highway DQN significantly outperforms all competitors in terms of both reward and sample efficiency on almost all the tasks. Especially on tasks like Seaquest, our Highway DQN achieves almost twice the reward of the best competitor. On tasks like Freeway, although several algorithms eventually converge to the level of Highway DQN, they require twice the samples needed by Highway DQN.

Ablation Study. We conducted the following ablation studies to investigate properties of Highway DQN. (1) Lookahead step ($\mathcal{N} = \{0, \cdots, N-1\}$ for Highway DQN and N for Multi-step DQN). As shown in Fig. 4a, compared to Multi-step DQN, our Highway DQN shows strong robustness against variations of the lookahead step. This is because Highway DQN can adaptively choose the lookahead step. (2) Replay buffer size. As shown in Fig. 4b, when the memory size increases to 8×10^5 (orange line), our Highway DQN shows a performance improvement, while Multi-Step DQN shows a degradation. (3) Softmax Temperature α and number of target networks. As shown in Fig. 5a and Fig. 5b, appropriate softmax temperature α and number of target networks can reduce the overestimation of Q values and benefit the performance of the algorithm.

8 Conclusions

We introduced a novel multi-step Bellman Optimality Equation for efficient multi-step credit assignment (CA) in reinforcement learning (RL). We proved that its solution is the optimal value function, and that the corresponding policy adjustments generally converge faster than the traditional Bellman Optimality operator. Our *Highway RL* methods combine the best of direct policy search (where CA is performed without trying to identify useful environmental states or subgoals during runtime), and standard RL, which finds useful states/subgoals through dynamic programming. Highway RL quickly and safely extracts useful sub-policies derived from data obtained through previously tested policies. The derived algorithms have several advantages over existing off-policy algorithms. Their feasibility and effectiveness were experimentally illustrated on a series of standard benchmark datasets. Future work will theoretically analyze the algorithm's behavior in the model-free case.

Ethics Statement

This paper provides a new operator for off-policy learning based methods. The authors do not find any particular concerns w.r.t the potential ethical or societal consequence.

REPRODUCIBILITY STATEMENT

The code is available at https://anonymous.4open.science/r/Highway-Reinforcement-Learning-4202 and more implementation details can be found at Appendix B.

REFERENCES

Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. Rudder: Return decomposition for delayed rewards. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Kristopher De Asis, J. Fernando Hernandez-Garcia, G. Zacharias Holland, and Richard S. Sutton. Multi-step reinforcement learning: A unifying algorithm. In *AAAI*, pp. 2902–2909, 2017.

André Barreto, Shaobo Hou, Diana Borsa, David Silver, and Doina Precup. Fast reinforcement learning with generalized policy updates. *Proceedings of the National Academy of Sciences*, 117 (48):30079–30087, 2020.

Gabriel Barth-Maron, Matthew W Hoffman, David Budden, Will Dabney, Dan Horgan, Dhruva Tb, Alistair Muldal, Nicolas Heess, and Timothy Lillicrap. Distributed distributional deterministic policy gradients. *arXiv preprint arXiv:1804.08617*, 2018.

Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1st edition, 1957.

Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for openai gym. https://github.com/maximecb/gym-minigrid, 2018.

Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. In *Advances in neural information processing systems*, pp. 442–450, 2010.

Yonathan Efroni, Gal Dalal, Bruno Scherrer, and Shie Mannor. Beyond the one-step greedy approach in reinforcement learning. In *International Conference on Machine Learning*, pp. 1387–1396. PMLR, 2018.

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

Roy Fox, Ari Pakman, and Naftali Tishby. Taming the noise in reinforcement learning via soft updates. *arXiv preprint arXiv:1512.08562*, 2015.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. *International Conference on Machine Learning (ICML)*, pp. 1352–1361, 2017.

Anna Harutyunyan, Marc G Bellemare, Tom Stepleton, and Rémi Munos. Q (λ) with off-policy corrections. In *International Conference on Algorithmic Learning Theory*, pp. 305–320. Springer, 2016.

Dan Horgan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado Van Hasselt, and David Silver. Distributed prioritized experience replay. *arXiv preprint arXiv:1803.00933*, 2018.

Qingfeng Lan. A pytorch reinforcement learning framework for exploring new ideas. https://github.com/qlan3/Explorer, 2019.

Qingfeng Lan, Yangchen Pan, Alona Fyshe, and Martha White. Maxmin q-learning: Controlling the estimation bias of q-learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Bkg0u3Etwr.

Alberto Maria Metelli, Matteo Papini, Francesco Faccio, and Marcello Restelli. Policy optimization via importance sampling. In *Advances in Neural Information Processing Systems*, pp. 5442–5454, 2018.

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc Bellemare. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 1054–1062, 2016.

Doina Precup. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, pp. 80, 2000.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

I. Rechenberg. Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Dissertation, 1971. Published 1973 by Fromman-Holzboog.

Mark Rowland, Will Dabney, and Rémi Munos. Adaptive trade-offs in off-policy learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 34–44. PMLR, 2020.

Sharma Sahil, Raguvir J Girish, Ramesh Srivatsan, and Ravindran Balaraman. Learning to mix n-step returns: Generalizing lambda-returns for deep reinforcement learning. *CoRR*, abs/1705.07445, 2017. URL http://arxiv.org/abs/1705.07445.

J. Schmidhuber. Learning to generate sub-goals for action sequences. In T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas (eds.), *Artificial Neural Networks*, pp. 967–972. Elsevier Science Publishers B.V., North-Holland, 1991.

J. Schmidhuber. Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4(2):234–242, 1992.

John Schulman, Philipp Moritz, Sergey Levine, Michael I Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *international conference on learning representations*, 2016.

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

Zhao Song, Ron Parr, and Lawrence Carin. Revisiting the softmax bellman operator: New benefits and new perspective. In *International conference on machine learning*, pp. 5916–5925. PMLR, 2019.

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

Csaba Szepesvári. Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning*, 4(1):1–103, 2010.

Manan Tomar, Yonathan Efroni, and Mohammad Ghavamzadeh. Multi-step greedy reinforcement learning algorithms. In *International Conference on Machine Learning*, pp. 9504–9513. PMLR, 2020.

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

Martha White and Adam M. White. A greedy approach to adapting the trace parameter for temporal difference learning. *CoRR*, abs/1607.00446, 2016. URL http://arxiv.org/abs/1607.00446.

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

Kenny Young and Tian Tian. Minatar: An atari-inspired testbed for more efficient reinforcement learning experiments. 2019. URL http://arxiv.org/abs/1903.03176.

A THEOREM PROOFS

A.1 PROOF OF PROPERTIES OF HIGHWAY OPERATOR $\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}$

Theorem 1 *(Highway Bellman Optimality Equation (Highway Equation)) Let us have an MDP $\mathcal{M}$. For any $\widehat{\Pi}$ and $\mathcal{N}$ (s.t. $0 \in \mathcal{N}$), we have*

1) $\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}$ is a contraction on complete metric space $l_\infty(\mathcal{S})$, i.e. for any $V, V' \in l_\infty(\mathcal{S})$, we have[6] $\|\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} V - \mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} V'\| \leq \|\mathcal{B}V - \mathcal{B}V'\| \leq \gamma \|V - V'\|$;

2) V^* is the only fixed point $\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}$. That is,

$$\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} V^* = V^*, \tag{13}$$

and for all $V \in l_\infty(S)$ holds $V = \mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} V$ if and only if $V = V^*$.

3) For any $V_0 \in l_\infty(\mathcal{S})$ and any sequence of set of behavioral policies $(\widehat{\Pi}_1, \cdots, \widehat{\Pi}_k)$, the sequence $(\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}_k} \circ \mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}_{k-1}} \circ \ldots \circ \mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}_1})[V_0]$,$k \in \mathbb{N}$ converge R-linearly to V^* with convergence rate γ.

Proof: 1. The contraction property can be obtained:

$$\begin{aligned}
\left\|\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} V - \mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} V'\right\| &= \left\|\max_{\pi \in \widehat{\Pi}} \max_{n \in \mathcal{N}} (\mathcal{B}^\pi)^n \mathcal{B}V - \max_{\pi \in \widehat{\Pi}} \max_{n \in \mathcal{N}} (\mathcal{B}^\pi)^n \mathcal{B}V'\right\| \\
&\leq \max_{\pi \in \widehat{\Pi}} \max_{n \in \mathcal{N}} \|(\mathcal{B}^\pi)^n \mathcal{B}V - (\mathcal{B}^\pi)^n \mathcal{B}V'\| \\
&\leq \max_{\pi \in \widehat{\Pi}} \max_{n \in \mathcal{N}} \gamma^n \|\mathcal{B}V - \mathcal{B}V'\| \\
&\leq \|\mathcal{B}V - \mathcal{B}V'\| \\
&\leq \gamma \|V - V'\|
\end{aligned}$$

2. By application of the Banach fixed point theorem to point 1., the operator $\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}$ must have a unique fixed point. Therefore, it suffices to verify that V^* is a fixed point of $\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}$. Our operator can be rewritten as

$$\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} V \triangleq \max_{\pi \in \widehat{\Pi}} \max_{n \in \mathcal{N}} (\mathcal{B}^\pi)^n \mathcal{B}V.$$

As $(\mathcal{B}^\pi)^n \mathcal{B}V^* \leq (\mathcal{B})^n V^* = V^*$ for any π and $n \geq 0$ (and the equivalence holds when $n = 0$), we get

$$\max_{\pi \in \widehat{\Pi}} \max_{n \in \mathcal{N}} (\mathcal{B}^\pi)^n \mathcal{B}V^* = V^*.$$

3. From 1) and 2) follows: $\|(\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}_k} \circ \mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}_{k-1}} \circ \ldots \mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}_1})[V_0] - V^*\| = \|(\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}_k} \circ \mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}_{k-1}} \circ \ldots \mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}_1})[V_0] - \mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}_k} V^*\| \leq \gamma \|(\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}_{k-1}} \circ \ldots \mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}_1})[V_0] - V^*\|$. By repeating the same argument we end up with $\|(\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}_k} \circ \mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}_{k-1}} \circ \ldots \mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}_1})[V_0] - V^*\| \leq \gamma^k \|V_0 - V^*\|$ from which the statement follows.

□

Theorem 6 *(Highway Bellman Optimality Equation (Highway Equation) w.r.t. action VF Q) Consider MDP $\mathcal{M}$. For any $\widehat{\Pi}$ and $\mathcal{N}$ (s.t. $0 \in \mathcal{N}$), we have*

1. $\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}$ is a contraction on complete metric space $l_\infty(\mathcal{S} \times \mathcal{A})$, i.e. for any $Q, Q' \in l_\infty(\mathcal{S} \times \mathcal{A})$, we have $\|\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} Q - \mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} Q'\| \leq \|\mathcal{B}Q - \mathcal{B}Q'\| \leq \gamma \|Q - Q'\|$;

2. Q^* is the only fixed point $\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}$. That is,

$$\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} Q^* = Q^*, \tag{14}$$

and for all $Q \in l_\infty(\mathcal{S} \times \mathcal{A})$ holds $Q = \mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} Q$ if and only if $Q = Q^*$.

[6] We denote by $\|\cdot\|$ for supremum norm throughout the paper.

Proof: Note that $\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}$ w.r.t action VF Q can also be represented by the Bellman Operators w.r.t Q, i.e.,

$$\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} Q \triangleq \max_{\pi} \max_{n} \left(\mathcal{B}^{\pi}\right)^{n} \mathcal{B} Q.$$

Except for the change of value spaces ($l_\infty(\mathcal{S} \times \mathcal{A})$ instead of $l_\infty(\mathcal{S})$) the proof is the same as for theorem 1). □

Theorem 2 *For any α, any $\widehat{\Pi}$, and any $\mathcal{N}$, $\widetilde{\mathcal{G}}_{\mathcal{N}}^{\widehat{\Pi}}$ is a contraction and $\widetilde{\mathcal{G}}_{\mathcal{N}}^{\widehat{\Pi}} Q^* = Q^*$ holds.*

Proof:

$$\widetilde{\mathcal{G}}_{\mathcal{N}}^{\widehat{\Pi}} Q\left(s_0, a_0\right) \triangleq \underset{\pi \in \widehat{\Pi}}{smax^{\alpha}} \underset{n' \in \mathcal{N}}{smax^{\alpha}} \max_{n \in \{0, n'\}} \mathbb{E}_{\tau_{s_0, a_0}^{n+1} \sim \pi}\left[G_Q^{n+1}(\tau_{s_0, a_0}^{n+1})\right], \tag{15}$$

This operator can be represented as

$$\widetilde{\mathcal{G}}_{\mathcal{N}}^{\widehat{\Pi}} Q \triangleq \underset{\pi \in \widehat{\Pi}}{smax^{\alpha}} \underset{n' \in \mathcal{N}}{smax^{\alpha}} \max_{n \in \{0, n'\}} \left(\mathcal{B}^{\pi}\right)^{n} \mathcal{B} Q.$$

First, following the proof in Theorem 1, we have $\left(\mathcal{B}^{\pi}\right)^{n} \mathcal{B} Q^* \leq \left(\mathcal{B}\right)^{n} Q^* = Q^*$, for any π and $n \geq 0$ (and the equivalence holds when $n = 0$). Then, we have

$$\underset{\pi \in \widehat{\Pi}}{smax^{\alpha}} \underset{n' \in \mathcal{N}}{smax^{\alpha}} \max_{n \in \{0, n'\}} \left(\mathcal{B}^{\pi}\right)^{n} \mathcal{B} Q^* = Q^*$$

Given an action VF Q, let us define two distribution over $\widehat{\Pi}$ and $\mathcal{N}$, denoted by $\mathcal{P}_{\widehat{\Pi}}^{s}$ and $\mathcal{P}_{\mathcal{N}}^{s}$.

$$\mathbb{E}_{\pi \sim \mathcal{P}_{\widehat{\Pi}}^{s}, n' \sim \mathcal{P}_{\mathcal{N}}^{s}} \max_{n \in \{0, n'\}} \left(\mathcal{B}^{\pi}\right)^{n} \mathcal{B} Q\left(s\right) = \underset{\pi \in \widehat{\Pi}}{smax^{\alpha}} \underset{n' \in \mathcal{N}}{smax^{\alpha}} \max_{n \in \{0, n'\}} \left(\mathcal{B}^{\pi}\right)^{n} \mathcal{B} Q\left(s\right)$$

$$\begin{aligned}
\left\|\widetilde{\mathcal{G}}_{\mathcal{N}}^{\widehat{\Pi}} Q - Q^*\right\| &= \left\|\underset{\pi \in \widehat{\Pi}}{smax^{\alpha}} \underset{n' \in \mathcal{N}}{smax^{\alpha}} \max_{n \in \{0, n'\}} \left(\mathcal{B}^{\pi}\right)^{n} \mathcal{B} Q - Q^*\right\| \\
&= \left\|\mathbb{E}_{\pi \sim \mathcal{P}_{\widehat{\Pi}}, n' \sim \mathcal{P}_{\mathcal{N}}} \max_{n \in \{0, n'\}} \left(\mathcal{B}^{\pi}\right)^{n} \mathcal{B} Q - \mathbb{E}_{\pi \sim \mathcal{P}_{\widehat{\Pi}}, n' \sim \mathcal{P}_{\mathcal{N}}} \max_{n \in \{0, n'\}} \left(\mathcal{B}^{\pi}\right)^{n} \mathcal{B} Q^*\right\| \\
&\leq \max_{\pi \in \widehat{\Pi}} \max_{n \in \mathcal{N}} \left\|\left(\mathcal{B}^{\pi}\right)^{n} \mathcal{B} Q - \left(\mathcal{B}^{\pi}\right)^{n} \mathcal{B} Q^*\right\| \\
&\leq \gamma \left\|Q - Q^*\right\|
\end{aligned}$$

□

Theorem 3 *(Comparison to Bellman Optimality Operator $\mathcal{B}$)*
1) $\left\|\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} V - V^*\right\| \leq \left\|\mathcal{B} V - V^*\right\|$;
2) Assume that $V_0 \leq V^*$: for any s we have $\left|\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} V_0\left(s\right) - V^*\left(s\right)\right| \leq \left|\mathcal{B} V_0\left(s\right) - V^*\left(s\right)\right|$; the inequality holds as long as there exists $\pi' \in \widehat{\Pi}$ such that $\arg\max_{n \in \mathcal{N}} \left(\mathcal{B}^{\pi'}\right)^{n} \mathcal{B} V_0\left(s\right) > 0$.

Proof: 1) has been proved in Theorem 1.

2) Note that $\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} V_0 \geq \mathcal{B} V_0$. If $V_0 \leq V^*$, then $\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} V_0 \leq V^*$ and $\mathcal{B} V_0 \leq V^*$ (as they are both monotonic). □

Theorem 4 *(Comparison to Multi-Step Bellman Expectation Operator) Assume that the VF $V_k \leq V^*$. Let $V_{k+1}^{\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}_k}}$ and $V_{k+1}^{\mathcal{B}_N^{\pi_k}}$ denote the $k+1$-th VF of Highway Value Iteration with hyperparameter $\mathcal{N}$ and Generalized Policy Iteration with hyperparameter N. We have*

$$\left\|V_{k+1}^{\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}_k}} - V^*\right\| \leq \min_{N \in \mathcal{N}} \left\|V_{k+1}^{\mathcal{B}_{N+1}^{\pi_k}} - V^*\right\| \tag{16}$$

Proof: First, $\max_{\pi\in\widehat{\Pi}}\max_{n\in\mathcal{N}}\left(\mathcal{B}^{\pi}\right)^{n}\mathcal{B}V_k\left(s\right)\geq\left(\mathcal{B}^{\pi'}\right)^{N}\mathcal{B}V_k\left(s\right)$ for any $N\in\mathcal{N}$, $\pi'\in\widehat{\Pi}$, and any $s\in\mathcal{S}$.

According to Algorithm 1, we know that $\pi_k\in\widehat{\Pi}_k$ Then, for any s we have

$$\begin{aligned}
&\left|\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}V_k\left(s\right)-V^*\left(s\right)\right|\\
&=\left|\max_{\pi\in\widehat{\Pi}}\max_{n\in\mathcal{N}}\left(\mathcal{B}^{\pi}\right)^{n}\mathcal{B}V_k\left(s\right)-V^*\left(s\right)\right|\\
&=V^*\left(s\right)-\max_{\pi\in\widehat{\Pi}}\max_{n\in\mathcal{N}}\left(\mathcal{B}^{\pi}\right)^{n}\mathcal{B}V_k\left(s\right)\\
&=\min_{\pi\in\widehat{\Pi}}\min_{n\in\mathcal{N}}\left(V^*\left(s\right)-\left(\mathcal{B}^{\pi}\right)^{n}\mathcal{B}V_k\left(s\right)\right)\\
&=\min_{\pi\in\widehat{\Pi}}\min_{n\in\mathcal{N}}\left|\left(\mathcal{B}^{\pi}\right)^{n}\mathcal{B}V_k\left(s\right)-V^*\left(s\right)\right|\\
&\leq\min_{n\in\mathcal{N}}\left|\left(\mathcal{B}^{\pi_k}\right)^{n}\mathcal{B}V_k\left(s\right)-V^*\left(s\right)\right|\\
&=\min_{N\in\mathcal{N}}\left|\left(\mathcal{B}^{\pi_k}\right)^{N+1}V_k\left(s\right)-V^*\left(s\right)\right|\\
&=\min_{N\in\mathcal{N}}\left|V_{k+1}^{\mathcal{B}_{N+1}^{\pi_k}}-V^*\left(s\right)\right|
\end{aligned}$$

The inequality still holds if we use the supremum norm.

□

Assumption A$(\widehat{\Pi},n)$ *Given set of behavioral policies $\widehat{\Pi}$, a lookahead step n. Let $\mathcal{S}_{s,n}^{\pi'}$ denote the set of all possible visited states by executing policy π' for n steps, starting from state s. We have: for each $s\in\mathcal{S}$ there exists at least one policy $\pi'\in\widehat{\Pi}$ such that $\pi'(s')=\pi^*(s')$ for any $s'\in\mathcal{S}_{s,n}^{\pi'}$ and π^* an optimal policy. Note the quantification order: $(\forall s\in\mathcal{S},\exists\pi'\in\widehat{\Pi},\forall s'\in\mathcal{S}_{s,n}^{\pi'}):\pi'(s')=\pi^*(s')$.*

Theorem 5 *(Better Contraction Rate) Assume $N-1\in\mathcal{N}$, $\widehat{\Pi}$ satisfies Assumption A$(\widehat{\Pi},N-1)$, and $V_0\in l_\infty(\mathcal{S})$ and $V_0\leq V^*$. the convergence rate of $\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}$ is γ^N, i.e., $\|\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}V-V^*\|\leq\gamma^N\|V-V^*\|$;*

Proof: 1) First, if $\widehat{\Pi}$ satisfies assumption Assumption A$(\widehat{\Pi},N-1)$, that is, starting from any s, there exists at least one $\pi\in\widehat{\Pi}$ executing the optimal actions for $N-1$ steps, then we have

$$\begin{aligned}
\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}V&=\max_{\pi\in\widehat{\Pi}}\max_{n'\in\mathcal{N}}\left(\mathcal{B}^{\pi}\right)^{n'}\mathcal{B}V\\
&\geq\max_{\pi\in\widehat{\Pi}}\left(\mathcal{B}^{\pi}\right)^{N-1}\mathcal{B}V\\
&\geq\left(\mathcal{B}^{\pi^*}\right)^{N-1}\mathcal{B}V.
\end{aligned}$$

Second, if $V\leq V^*$, then $\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}V\leq V^*$ ($\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}$ is monotonic) and $\left(\mathcal{B}^{\pi^*}\right)^{N-1}\mathcal{B}V\leq V^*$.

Finally, with the results above, we have

$$\begin{aligned}
\left\|\mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}}V-V^*\right\|&\leq\left\|\left(\mathcal{B}^{\pi^*}\right)^{N-1}\mathcal{B}V-V^*\right\|\\
&=\left\|\left(\mathcal{B}^{\pi^*}\right)^{N-1}\mathcal{B}V-\left(\mathcal{B}^{\pi^*}\right)^{N-1}\mathcal{B}V^*\right\|\\
&\leq\gamma^N\left\|V-V^*\right\|.
\end{aligned}$$

□

To utilize the better convergence rates obtained in theorem 5, we have to be a bit more careful than in theorem 1 point *3)* and take care of the monotony requirements in theorem 5. This is done in the following lemma.

Lemma 1 *Assume a sequence of operators T_k, $k \in \mathbb{N}_0$ on $L_\infty(\mathcal{S})$, where all T_k satisfy $(\forall V \in L_\infty(\mathcal{S}), V \leq V^*) : \|T_k V - V^*\| \leq \gamma' \|V - V^*\|$ with common convergence rate γ' and common limit V^*. Further assume all T_k are monotonic and have fixed point V^*. Then for any $V_0 \in l_\infty(\mathcal{S}), V_0 \leq V^*$ the sequence $(T_k \circ T_{k-1} \circ \ldots T_0)[V_0]$,$k \in \mathbb{N}_0$ converges R-linearly to V^* with convergence rate γ'.*

Proof: Assuming $V_0 \in l_\infty(\mathcal{S}), V_0 \leq V^*$ and monotony and fixed point of operators $T_0, \ldots, T_{k-1}$ implies: $V := (T_{k-1} \circ T_{k-2} \circ \ldots T_0)[V_0] \leq V^*$. Now we can apply the inequality with γ' to get $\|(T_k \circ T_{k-1} \circ \ldots T_0)[V_0] - V^*\| \leq \gamma' \|(T_{k-1} \circ \ldots T_0)[V_0] - V^*\|$. By repeating the same argument we end up with $\|(T_k \circ T_{k-1} \circ \ldots T_0)[V_0] - V^*\| \leq \gamma'^k \|V_0 - V^*\|$ from which the statement follows.

□

Let us denote $\hat{\Pi}_k$ ($k \in \mathbb{N}$) the sequence of behavioral policies generated by algorithm 1. In Algorithm 1 the set of behavioral policies $\hat{\Pi}_k$ changes by adding a greedy policy every K iterations. Regardless of the sets $\hat{\Pi}_k$ the algorithm converges to the optimal value function form theorem 1 point *3)*. This means that more and more close to optimal policies are added to $\hat{\Pi}_k$ set with growing iteration number k. When assumption $A(\cdot, n)$ gets satisfied for some n at iteration k_0, then due to monotony of the sequence $\hat{\Pi}_k$,$k \in \mathbb{N}$ (i.e. the set $\hat{\Pi}_k$ can just grow over time) $A(\cdot, n)$ is satisfied for the whole suffix $\hat{\Pi}_k$,$k \geq k_0$. We can then use lemma 1 on corresponding suffix of operator sequence $\mathcal{G}_{\mathcal{N}}^{\hat{\Pi}_k}$,$k \geq k_0$ with $\gamma' = \gamma^{n+1}$ to claim better convergence rate (γ^{n+1}) of the corresponding suffix of value functions V_k,$k \geq k_0$. When $\mathcal{S}$ is finite this gives us monotonic improvement of the convergence rate of the sequence V_k,$k \in \mathbb{N}$ as $A(\cdot, n)$ gets eventually satisfied for bigger and bigger n.

Lemma 2 *If $V_0 \leq V^*$, then $(\mathcal{G}_{\mathcal{N}}^{\hat{\Pi}})^{\circ k} V_0 \leq V^*$ for any k.*

Proof: As $\mathcal{G}_{\mathcal{N}}^{\hat{\Pi}}$ is monotonic, we have $(\mathcal{G}_{\mathcal{N}}^{\hat{\Pi}}) V_0 \leq (\mathcal{G}_{\mathcal{N}}^{\hat{\Pi}}) V^* = V^*$. Applying $\mathcal{G}_{\mathcal{N}}^{\hat{\Pi}}$ again to the both side $(\mathcal{G}_{\mathcal{N}}^{\hat{\Pi}})^2 V_0 \leq (\mathcal{G}_{\mathcal{N}}^{\hat{\Pi}}) V^* = V^*$. Finally, we get $(\mathcal{G}_{\mathcal{N}}^{\hat{\Pi}})^k V_0 \leq (\mathcal{G}_{\mathcal{N}}^{\hat{\Pi}}) V^* = V^*$.

□

A.2 Proof of properties of multi-step optimality operator $\mathcal{B}_N^{\hat{\Pi}}$

Here we prove the theorems regarding multi-step optimality operator $\mathcal{B}_N^{\hat{\Pi}}$ ($N \geq 0$).

$$\mathcal{B}_N^{\hat{\Pi}} Q(s_0, a_0) \triangleq \mathbb{E}_{\pi \sim \mathcal{P}, \tau_{s_0, a_0}^N \sim \pi} \left[\sum_{t=0}^{N-1} \gamma^t r_t + \gamma^N \max_{a'_N} Q(s_N, a'_N) \right] \tag{17}$$

Lemma 3 *For any two vectors $Q, Q' \in l_\infty(\mathcal{S} \times \mathcal{A})$, $N \geq 0$, $\|\mathcal{B}_N^{\hat{\Pi}} Q - \mathcal{B}_N^{\hat{\Pi}} Q'\| \leq \gamma^N \|Q - Q'\|$.*

Proof:

$$\begin{aligned} &\|\mathcal{B}_N^{\hat{\Pi}} Q - \mathcal{B}_N^{\hat{\Pi}} Q'\| \leq \mathbb{E}_{\pi \sim \mathcal{P}} \left\| (\mathcal{B}^\pi)^{N-1} \mathcal{B} Q - (\mathcal{B}^\pi)^{N-1} \widetilde{\mathcal{B}} Q' \right\| \\ &\leq \gamma^N \|Q - Q'\| \end{aligned}$$

□

Theorem 7 *(the fixed point of multi-step optimality operator $\mathcal{B}_N^{\hat{\Pi}}$) Let $Q^*_{\mathcal{B}_N^{\hat{\Pi}}}$ denote the fixed point of $\mathcal{B}_N^{\hat{\Pi}}$, i.e., $\mathcal{B}_N^{\hat{\Pi}} Q^*_{\mathcal{B}_N^{\hat{\Pi}}} = Q^*_{\mathcal{B}_N^{\hat{\Pi}}}$. For any $N \geq 2$,*
1) $Q^*_{\mathcal{B}_N^{\hat{\Pi}}} \leq Q^*$;
2) $Q^*_{\mathcal{B}_N^{\hat{\Pi}}} = Q^*$ if and only if any $\pi \in \hat{\Pi}$, $\mathcal{P}(\pi) > 0$ satisfies $\pi(s) = \pi^*(s)$ for any $s \in U := \{s_1 \in \mathcal{S} | \exists s_0, a_0 \mathcal{T}(s_1|s_0, a_0) > 0\}$ and π^* an optimal policy.

Proof:

Applying Banach's fixed point theorem to $\mathcal{B}_N^{\widehat{\Pi}}$ using Lemma 3, we know that $\mathcal{B}_N^{\widehat{\Pi}}$ has only one fixed point.

1) For any $\pi \in \Pi$ and $N \geq 2$, we have $(\mathcal{B}^\pi)^{N-1}\mathcal{B}Q^* \leq (\mathcal{B})^{N-1}\mathcal{B}Q^* = Q^*$. This implies that

$$Q^* \geq \mathbb{E}_\pi(\mathcal{B}^\pi)^{N-1}\mathcal{B}Q^* = \mathcal{B}_N^{\widehat{\Pi}}Q^*. \tag{18}$$

Further, $\mathcal{B}_N^{\widehat{\Pi}}$ is monotonic, i.e., $Q \leq Q'$ implies $\mathcal{B}_N^{\widehat{\Pi}}Q \leq \mathcal{B}_N^{\widehat{\Pi}}Q'$. This implies that

$$Q^* \geq \mathcal{B}_N^{\widehat{\Pi}}Q^* \geq (\mathcal{B}_N^{\widehat{\Pi}})^2Q^* \geq \cdots \geq Q^*_{\mathcal{B}_N^{\widehat{\Pi}}}. \tag{19}$$

2) For the implication "$\Longleftarrow$"it suffices to show for all $\pi \in \widehat{\Pi}$, $\mathcal{P}(\pi) > 0$ $\mathcal{B}^\pi Q^* = Q^*$. Since then we get

$$\mathcal{B}_N^{\widehat{\Pi}}Q^* = \mathbb{E}_{\pi\sim\mathcal{P}}(\mathcal{B}^\pi)^{N-1}Q^* = \mathbb{E}_{\pi\sim\mathcal{P}}Q^* = Q^*$$

and the implication is proved. Thus let us fix $\pi \in \widehat{\Pi}$, $\mathcal{P}(\pi) > 0$ and $s_0 \in \mathcal{S}, a_0 \in \mathcal{A}$ the following holds $\mathcal{B}^\pi(s_0, a_0) = \mathbb{E}_{s_1\sim\mathcal{T}(\cdot|s_0,a_0)}\mathbb{E}_{a_1\sim\pi(\cdot|s_1)}[r(s_0, a_0)+\gamma Q^*(s_1, a_1)]$ Since In the first expectation we care just about s_1 for which $\mathcal{T}(s_1|s_0, a_0) > 0$ but then $s_1 \in U$. Since π on U can be replaced by the optimal policy π^* from the assumption we get

$$\begin{aligned}\mathcal{B}^\pi(s_0, a_0) &= \mathbb{E}_{s_1\sim\mathcal{T}(\cdot|s_0,a_0)}\mathbb{E}_{a_1\sim\pi(\cdot|s_1)}[r(s_0, a_0) + \gamma Q^*(s_1, a_1)]\\ &= \mathbb{E}_{s_1\sim\mathcal{T}(\cdot|s_0,a_0)}\mathbb{E}_{a_1\sim\pi^*(\cdot|s_1)}[r(s_0, a_0) + \gamma Q^*(s_1, a_1)] = Q^*(s_0, a_0).\end{aligned}$$

The remaining implication "$\Longrightarrow$" will be proved by contradiction. Thus assume the conclusion does not hold, i.e. there exist $\pi \in \widehat{\Pi}$, $\mathcal{P}(\pi) > 0$ and $s_1 \in U$ such that $\mathcal{P}(\pi) > 0$ and $\pi(\cdot|s_1)$ is not optimal. Since $s_1 \in U$ we there exists $s_0 \in \mathcal{S}, a_0 \in \mathcal{A}$ such that $\mathcal{T}(s_1|s_0, a_0) > 0$ First we aim to prove inequality

$$\mathcal{B}^\pi Q^*(s_0, a_0) < Q^*(s_0, a_0).$$

Since $\pi(\cdot|s_1)$ assigns positive probability to non-optimal action it is easy to reason (especially for finite $\mathcal{A}$) that

$$\mathbb{E}_{a_1\sim\pi(\cdot|s_1)}Q^*(s_1, a_1) < V^*(s_1).$$

For other states different from s_1 we can have still equality but anyway the countable sum leaves the inequality strict

$$\begin{aligned}(\mathcal{B}^\pi Q^*)(s_0, a_0) &= r(s_0, a_0) + \mathbb{E}_{s_1'\sim\mathcal{T}(\cdot|s_0,a_0)}\mathbb{E}_{a_1\sim\pi(\cdot|s_1')}Q^*(s_1', a_1)\\ &< r(s_0, a_0) + \mathbb{E}_{s_1'\sim\mathcal{T}(\cdot|s_0,a_0)}V^*(s_1') = Q^*(s_0, a_0).\end{aligned}$$

Now since $(\mathcal{B}^\pi)^{N-2}Q^* \leq Q^*$ we get

$$(\mathcal{B}^\pi)^{N-1}\mathcal{B}Q^* = (\mathcal{B}^\pi)^{N-1}Q^* = \mathcal{B}^\pi(\mathcal{B}^\pi)^{N-2}Q^* \leq \mathcal{B}^\pi Q^*.$$

Using previous result we obtain

$$((\mathcal{B}^\pi)^{N-1}\mathcal{B}Q^*)(s_0, a_0) \leq \mathcal{B}^\pi Q*(s_0, a_0) < Q^*(s_0, a_0).$$

and applying expectation with finite set $\widehat{\Pi}$ using $\mathcal{P}(\pi) > 0$ we get

$$\mathcal{B}_N^{\widehat{\Pi}}Q^*(s_0, a_0) < Q^*(s_0, a_0).$$

Further using the fact $\mathcal{B}_N^{\widehat{\Pi}}Q^* \leq Q^*$ and monotony of $\mathcal{B}_N^{\widehat{\Pi}}$ we get monotonic sequence $(\mathcal{B}_N^{\widehat{\Pi}})^kQ^*$,$k \in \mathbb{N}$ which eventually converges to its fixed point $(\mathcal{B}_N^{\widehat{\Pi}})^kQ^* \searrow Q^*_{\mathcal{B}_N^{\widehat{\Pi}}}$ which with previous result gives

$$Q^*_{\mathcal{B}_N^{\widehat{\Pi}}}(s_0, a_0) < Q^*(s_0, a_0).$$

□

B METHOD

B.1 ALGORITHMS

B.1.1 HIGHWAY VALUE ITERATION

The Highway Q-Learning algorithm is presented as Algorithm 1.

B.1.2 HIGHWAY Q LEARNING

The Highway Q learning algorithm is presented as Algorithm 3.

B.1.3 HIGHWAY DQN

The Highway DQN algorithm is presented as Algorithm 4.

C EXPERIMENTAL RESULTS

The code of this paper is publicly-available at `https://anonymous.4open.science/r/Highway-Reinforcement-Learning-4202`.

C.1 EXPERIMENTS WITH MODEL-BASED ALGORITHMS

C.1.1 DETAILS OF ENVIRONMENTS

Multi-Room is a grid world environment with multiple rooms connected by doors. The agent's goal is to reach a goal square in the opposite corner and get a reward ($r = 1000$). In addition, the agent will get a small reward $r = 0.001$ when it finds the exit door of the room. We use the implementations based on the gym-minigrid (Chevalier-Boisvert et al., 2018).

C.2 EXPERIMENTS OF MODEL-FREE ALGORITHMS IN TOY TASKS

Here we present the details of model-free-algorithms in artificial tasks. We adopt the experimental setting of Arjona-Medina et al. (2019).

C.2.1 DETAILS OF ENVIRONMENTS

We evaluate the model-free algorithms on two toy tasks involving delayed rewards (Arjona-Medina et al., 2019), where a reward is only provided at the end of each trial and is associated with the previous actions. For example, in task "Trace Back," the final reward depends on the first two actions. Each task is run with 100 random seeds. In task "Choice," the stochastic reward depends

Algorithm 1 Highway Value Iteration

Input: Initial set of behavioral policies $\widehat{\Pi}_0$; the set of lookahead steps $\mathcal{N}$; interval for adding new policy K .
Initialize: Initial VF $V_0 \in \mathbb{R}^{|\mathcal{S}|}$, ϵ, $k = 0$.
repeat
  if k mod K == 0 **then**
    $\pi_k(s) = \arg\max_a [r(s,a) + \gamma \mathbb{E}_{s'}[V_k(s')]]$
    $\widehat{\Pi}_k = \widehat{\Pi}_k \cup \{\pi_k\}$
  end if
  $V_{k+1} \leftarrow \mathcal{G}_{\mathcal{N}}^{\widehat{\Pi}} V_k$
  $k \leftarrow k + 1$
  $\widehat{\Pi}_k \leftarrow \widehat{\Pi}_{k-1}$
until $\|V_k - V_{k-1}\|_\infty \leq \epsilon$
Output: V_k

Algorithm 2 Generalized Policy Iteration

Input: the lookahead step N; interval for adding new policy K .
Initialize: Initial VF $V_0 \in \mathbb{R}^{|\mathcal{S}|}$, ϵ, $k = 0$.
repeat
  $\pi_k(s) = \arg\max_a [r(s,a) + \gamma \mathbb{E}_{s'}[V_k(s')]]$
  $V_{k+1} \leftarrow (\mathcal{B}^{\pi_k})^{\circ N} V_k$
  $k \leftarrow k + 1$
until $\|V_k - V_{k-1}\|_\infty \leq \epsilon$
Output: V_k

Algorithm 3 Highway Q learning

Input: Set of lookahead steps $\mathcal{N}$; Epochs of running algorithm I_{run}; Epochs of rolling-out policy I_{rollout}; Epochs of updating value function I_{update}; Initial value function $Q_0 \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}$; Exploration rate ϵ.
Initialize: $k = 0$; State-action replay buffer $\mathcal{D} = \emptyset$; .
for $m = 1, \cdots, I_{\text{run}}$ **do**
  Set π_m to be ϵ-greedy policy with Q_k
  $\mathcal{D}_{s,a}^{(m)} \leftarrow \emptyset$, for all $(s,a) \in \mathcal{S} \times \mathcal{A}$
  for $j = 1, \cdots, I_{\text{rollout}}$ **do**
    Collect a trajectory $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \cdots, s_T)$ with π_m.
    for $t = 0, 1, \cdots, T-1$ **do**
      Add $(s_t, a_t, r_t, s_{t+1}, a_{t+1}, r_{t+1}, \cdots, s_T)$ to $\mathcal{D}_{s_t,a_t}^{(m)}$
      Add (s_t, a_t) to $\mathcal{D}$
    end for
  end for
  for $j = 1, \cdots, I_{\text{update}}$ **do**
    Sample a (s_0, a_0) from $\mathcal{D}$.
    Update the value function by the following rule

$$Q_{k+1}(s_0, a_0) = \max_{m \in \mathsf{M}_{s_0,a_0}} \max_{n \in \mathcal{N}} \widehat{\mathbb{E}}^{\mathcal{D}_{s_0,a_0}^{(m)}} \left[G_{Q_k}^{n+1}(\tau_{s_0,a_0}^{n+1}) \right]$$

$$\text{where} \quad \widehat{\mathbb{E}}^{\mathcal{D}_{s_0,a_0}^{(m)}}[\cdot] = \frac{1}{\mathcal{D}_{s,a}^{(m)}} \sum_{r_{s_0,a_0}^{n+1} \in \mathcal{D}_{s_0,a_0}^{(m)}} [\cdot]$$

$$\mathsf{M}_{s,a} = \left\{ m \middle| |\mathcal{D}_{s,a}^{(m)}| \neq 0 \right\}$$

    $k = k + 1$
  end for
end for

on the first action at the beginning; the final reward depends on the first two actions. Please refer to Arjona-Medina et al. (2019) for more details.

C.2.2 Algorithmic Details

The following methods are compared:

- RUDDER with reward redistribution for Q-value estimation, and RUDDER applied on top of Q-learning.
- Q-learning with eligibility traces according to Watkins ($Q(\lambda)$).
- SARSA with eligibility traces (SARSA(λ)).
- Monte Carlo.

For RUDDER, we use the default setting of Arjona-Medina et al. (2019). For Q(λ) and SARSA(λ), the hyperparameter of eligibility traces is $\lambda = 0.9$. For Q(λ), we use Watkins' implementation.

Algorithm 4 Highway DQN

Input: Set of lookahead steps $\mathcal{N}$; Epochs of running algorithm I_{run}; Epochs of rolling-out policy I_{rollout}; Epochs of updating value function I_{update}; Batch size of updating value function b; Initial parameter θ of Q function; Exploration rate ϵ; Temperature hyperparameter α for softmax function $smax^{\alpha}$.
Initialize: State-action replay buffer $\mathcal{D} = \emptyset$; Initialize parameter θ' of target Q function, $\theta' \leftarrow \theta$.
for $m = 1, \cdots, I_{\text{run}}$ **do**
  Set π_m to be ϵ-greedy policy with Q_θ
  $\mathcal{D}_{s,a}^{(m)} \leftarrow \emptyset$, for all $(s,a) \in \mathcal{S} \times \mathcal{A}$
  for $j = 1, \cdots, I_{\text{rollout}}$ **do**
    Collect a trajectory $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \cdots, s_T)$ with π_m
    for $t = 0, 1, \cdots, T-1$ **do**
      Add $(s_t, a_t, r_t, s_{t+1}, a_{t+1}, r_{t+1}, s_{t+2}, a_{t+2}, \cdots, s_T)$ to $\mathcal{D}_{s_t,a_t}^{(m)}$
      Add (s_t, a_t) to $\mathcal{D}$
    end for
  end for
  for $j = 1, \cdots, I_{\text{update}}$ **do**
    Sample a mini-batch $\widehat{\mathcal{D}} = \{(s,a)\}$ from $\mathcal{D}$.
    Update θ by minimizing the following loss function:

$$L(\theta) = \sum_{(s,a)\in\widehat{\mathcal{D}}} \left[Q_\theta(s,a) - \underset{m\in \mathsf{M}_{s,a}}{smax^{\alpha}} \underset{n'\in\mathcal{N}}{smax^{\alpha}} \max_{n\in\{0,n'\}} \widehat{\mathbb{E}}^{\mathcal{D}_{s,a}^{(m)}} \left[G_{Q_{\theta'}}^{n+1}(\tau_{s,a}^{n+1}) \right] \right]^2 \tag{20}$$

  end for
  $\theta' \leftarrow \theta$ every C timesteps
end for

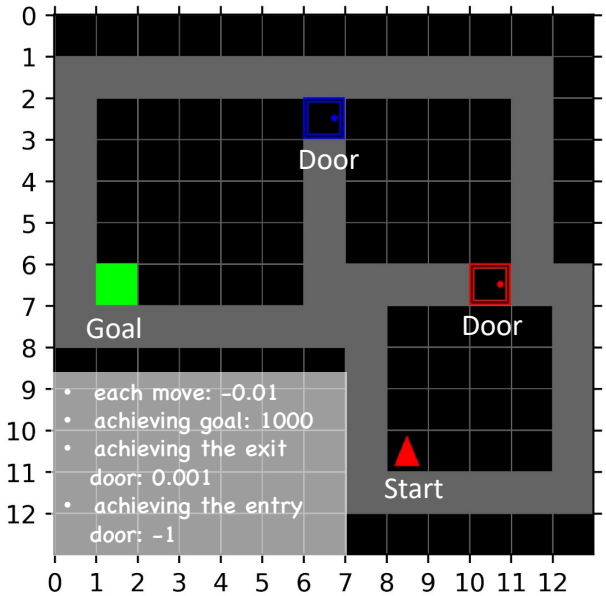


Figure 6: The illustration of the Minimalistic Gridworld Environment

The algorithms are evaluated until the task is solved. For MC, Q-values are the exponential moving average of the episode return. In all experiments, an ϵ-greedy policy with $\epsilon = 0.2$ is adopted.

The set of lookahead steps $\mathcal{N}$ is set by $\{0, 1, 2, \cdots, N\}$, where $N = T - t$ is set to be the length until the end (T is the length of trajectory and t the timestep of current state). Epochs of rolling-out policy is set by $I_{\text{rollout}} = 1$. Epochs of updating value function is set by $I_{\text{update}} = T$.

C.3 Experiments of Model-free Algorithms in MinAtar Tasks

C.3.1 Environmental Details

This paper uses 5 games in MinAtar (Young & Tian, 2019), including Asterix, Breakout, Freeway, Seaquest, and Space Invaders. The details of the environments can be found at `https://github.com/kenjyoung/MinAtar`.

C.3.2 Details of Algorithms

Hyperparameter	Value
Optimizer	RMSprop
Batch size	32
Gradient Clipping	Gym:5 MinAtari:1
Target network update frequency	Gym:200 MinAtari:1000
Hidden units of Q network	Gym: Fully-connected layer (64,64) MinAtari: Conv. layer(out channels=16, kernel size=3, stride=1) Conv. layer(out channels=16, kernel size=3, stride=1)
Activation function	ReLU
Buffer size	Gym:10^4 MinAtari:10^5
Discount factor γ	0.99
Exploration rate ϵ	statr:1.0; end: 0.1; decay type: linear

Table 2: Hyperparameters of the implemented algorithms. We reused hyperparameters and settings of neural networks in the Maxmin DQN paper (Lan et al., 2020).

We implement our and competing methods by extending the implementation of Maxmin DQN Lan (2019). The hyperparameters of training deep neural network are provided in Table 2.

We reuse the hyper-parameters and settings of neural networks in the Maxmin DQN paper (Lan et al., 2020). For Maxmin DQN, the best number of target networks (a hyperparameter) was chosen from $[2, 3, 4, 5, 6, 7, 8, 9]$ and the best learning rates were chosen from $[3 \times 10^{-3}; 10^{-3}; 3 \times 10^{-4}; 10^{-4}; 3 \times 10^{-5}]$. For our Highway DQN, Multi-Step DQN and Multi-Step SARSA, the best maximal lookahead step was chosen from $[4, 8, 12]$; the best learning rates from $[3\times10^{-3}; 3\times10^{-4}]$, and the best number of target networks was chosen from $[2, 4, 6]$. For our Highway DQN, the best hyperparameter of softmax temperature α is chosen from $\{0.1, 0.5\}$. Epochs of rolling-out policy is set by $I_{\text{rollout}} = 1$.

In practice, we adopt several measures to accelerate the computation. We cache the Q values generated by the target network for data (s, a) such that they can be reused when the data is sampled again for training until the target network is updated. With the cached Q values, all we need is a softmax over $|\mathcal{N}| \times |\mathcal{K}|$ numbers, which is typically fast on GPUs.