
Self-Imitation Learning via Generalized Lower Bound Q-learning

Yunhao Tang
Columbia University
yt2541@columbia.edu

Abstract

Self-imitation learning motivated by lower-bound Q-learning is a novel and effective approach for off-policy learning. In this work, we propose a n -step lower bound which generalizes the original return-based lower-bound Q-learning, and introduce a new family of self-imitation learning algorithms. To provide a formal motivation for the potential performance gains provided by self-imitation learning, we show that n -step lower bound Q-learning achieves a trade-off between fixed point bias and contraction rate, drawing close connections to the popular uncorrected n -step Q-learning. We finally show that n -step lower bound Q-learning is a more robust alternative to return-based self-imitation learning and uncorrected n -step, over a wide range of continuous control benchmark tasks.

1 Introduction

Learning with off-policy data is of central importance to scalable reinforcement learning (RL). The traditional framework of off-policy learning is based on importance sampling (IS): for example, in policy evaluation, given trajectories $(x_t, a_t, r_t)_{t=0}^{\infty}$ generated under behavior policy μ , the objective is to evaluate Q-function $Q^{\pi}(x_0, a_0)$ of a target policy π . Naive IS estimator involves products of the form $\pi(a_t | x_t) / \mu(a_t | x_t)$ and is infeasible in practice due to high variance. To control the variance, a line of prior work has focused on operator-based estimation to avoid full IS products, which reduces the estimation procedure into repeated iterations of off-policy evaluation operators [1–3]. Each iteration of the operator requires only local IS ratios, which greatly stabilizes the update.

More formally, such operators \mathcal{T} are designed such that their fixed points are the target Q-function $\mathcal{T}Q^{\pi} = Q^{\pi}$. As such, these operators are *unbiased* and conducive to theoretical analysis. However, a large number of prior work has observed that certain *biased* operators tend to have significant empirical advantages [4–6]. One notable example is the uncorrected n -step operator, which directly bootstraps from n -step target trajectories without IS corrections [4]. The removal of all IS ratios biases the estimate, but allows the learning signal to be propagated over a longer horizon (in Section 2, we will characterize such effects as contraction rates). Indeed, when behavior trajectories are unlikely under the current policy, small IS ratios $\pi(a_t | x_t) / \mu(a_t | x_t)$ quickly cut off the learning signal. In general, there is a trade-off between the fixed point bias and contraction rates. Empirical findings suggest that it might be desirable to introduce bias in exchange for faster contractions in practice [7].

Recently, self-imitation learning (SIL) has been developed as a family of novel off-policy algorithms which facilitate efficient learning from highly off-policy data [8–10]. In its original form, SIL is motivated as lower bound Q-learning [11]. In particular, let $Q_L(x, a) \leq Q^{\pi^*}(x, a)$ denote a lower bound of the optimal Q-function Q^{π^*} . Optimizing auxiliary losses which encourage $Q_{\theta}(x, a) \geq Q_L(x, a)$ could significantly speed up learning with the trained Q-function $Q_{\theta}(x, a)$. Such auxiliary losses could be extended to actor-critic algorithms with stochastic policies [8]: SIL suggests optimizing a policy $\pi_{\theta}(a | x)$ by maximizing an objective similar to $[Q^{\mu}(a | x) - V^{\pi_{\theta}}(x)]_+ \log \pi_{\theta}(a | x)$, where $V^{\pi_{\theta}}(x)$ is the value-function for policy π_{θ} , with $[x]_+ := \max(0, x)$. The update is intuitively

reasonable: if a certain actions a is high-performing under behavior policy μ , such that $Q^\mu(x, a) > V^{\pi_\theta}(x)$, the policy $\pi_\theta(a | x)$ should imitate such actions.

On a high-level, SIL is similar to the uncorrected n -step update in several aspects. With no explicit IS ratios, both methods entail that off-policy learning signals propagate over long horizons without being *cut-off*. As a result, both methods are biased due to the absence of proper corrections, and could be seen as trading-off fixed point bias for fast contractions.

Main idea. In this paper, we make several theoretical and empirical contributions.

- **Generalized SIL.** In Section 3, we propose generalized SIL which strictly extends the original SIL formulation [8]. Generalized SIL provides additional flexibility and advantages over the original SIL: it learns from partial trajectories and bootstraps with learned Q-function; it applies to both stochastic and deterministic actor-critic algorithms.
- **Trade-off.** In Section 4, we formalize the trade-offs of SIL. We show that generalized SIL trades-off contraction rates with fixed point bias in a similar way to uncorrected n -step [7]. Unlike uncorrected n -step, for which fixed point bias could be either positive or negative, the operator for SIL induces *positive* bias, which fits the motivation of SIL to move towards optimal Q-functions.
- **Empirical.** In Section 5, we show generalized SIL outperforms alternative baseline algorithms.

2 Background

Consider the standard formulation of markov decision process (MDP). At a discrete time $t \geq 0$, an agent is in state $x_t \in \mathcal{X}$, takes action $a_t \in \mathcal{A}$, receives a reward $r_t = r(x_t, a_t) \in \mathbb{R}$ and transitions to a next state $x_{t+1} \sim p(\cdot | x_t, a_t) \in \mathcal{X}$. A policy $\pi(a | x) : \mathcal{X} \mapsto \mathcal{P}(\mathcal{A})$ defines a map from state to distributions over actions. The standard objective of RL is to maximize the expected cumulative discounted returns $J(\pi) := \mathbb{E}_\pi[\sum_{t \geq 0} \gamma^t r_t]$ with a discount factor $\gamma \in (0, 1)$.

Let $Q^\pi(x, a)$ denote the Q-function under policy π and $Q^\pi \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{A}|}$ its vector form. Denote the Bellman operator as \mathcal{T}^π and optimality operator as \mathcal{T}^* [12]. Let π^* be the optimal policy, i.e. $\pi^* = \arg \max_\pi J(\pi)$. It follows that Q^π, Q^{π^*} are the unique fixed points of $\mathcal{T}^\pi, \mathcal{T}^*$ respectively [13]. Popular RL algorithms are primarily motivated by the fixed point properties of the Q-functions (or value functions): in general, given a parameterized Q-function $Q_\theta(x, a)$, the algorithms proceed by minimizing an empirical Bellman error loss $\min_\theta \mathbb{E}_{(x,a)} [(Q_\theta(x, a) - \mathcal{T}Q_\theta(x, a))^2]$ with operator \mathcal{T} . Algorithms differ in the distribution over sampled (x, a) and the operator \mathcal{T} . For example, Q-learning sets the operator $\mathcal{T} = \mathcal{T}^*$ for value iteration and the samples (x, a) come from an experience replay buffer [14]; Actor-critic algorithms set the operator $\mathcal{T} = \mathcal{T}^{\pi_\theta}$ for policy iteration and iteratively update the policy π_θ for improvement, the data (x, a) could be either on-policy or off-policy [15–18].

2.1 Elements of trade-offs in Off-policy Reinforcement Learning

Here we introduce elements essential to characterizing the trade-offs of generic operators \mathcal{T} in off-policy RL. For a complete review, please see [7]. Take off-policy evaluation as an example: the data are generated under a behavior policy μ while the target is to evaluate Q^π . Consider a generic operator \mathcal{T} and assume that it has fixed point \tilde{Q} . Define the contraction rate of the operator as $\Gamma(\mathcal{T}) := \sup_{Q_1 \neq Q_2} \|\mathcal{T}(Q_1 - Q_2)\|_\infty / \|Q_1 - Q_2\|_\infty$. Intuitively, operators with small contraction rate should have *fast* contractions to the fixed point. In practical algorithms, the quantity $\mathcal{T}Q(x, a)$ is approximated via stochastic estimations, denoted as $\tilde{\mathcal{T}}Q(x, a)$. All the above allows us to define the bias and variance of an operator $\mathbb{B}(\mathcal{T}) := \|\tilde{Q} - Q^\pi\|_2^2, \mathbb{V}(\mathcal{T}) := \mathbb{E}_\mu[\|\tilde{\mathcal{T}}Q - \mathcal{T}Q\|_2^2]$ evaluated at a Q-function Q . Note that all these quantities depend on the underlying MDP M , though when the context is clear we omit the notation dependency.

Ideally, we seek an operator \mathcal{T} with small bias, small variance and small contraction rate. However, it follows that these three aspects could not be optimized simultaneously for a general class of MDPs $M \in \mathcal{M}$

$$\sup_{M \in \mathcal{M}} \{\mathbb{B}(\mathcal{T}) + \sqrt{\mathbb{V}(\mathcal{T})} + \frac{2r_{\max}}{1-\gamma} \Gamma(\mathcal{T})\} \geq I(\mathcal{M}), \quad (1)$$

where $r_{\max} := \max_{x,a} r(x, a)$ and $I(\mathcal{M})$ is a information-theoretic lower bound [7]. This inequality characterizes the fundamental trade-offs of these three quantities in off-policy learning. Importantly,

we note that though the variance $\mathbb{V}(\mathcal{T})$ is part of the trade-off, it is often not a major focus of algorithmic designs [4, 7]. We speculate it is partly because in practice the variance could be reduced via e.g. large training batch sizes, while the bias and contraction rates do not improve with similar techniques. As a result, henceforth we focus on the trade-off between the bias and contraction rate.

2.2 Trading off bias and contraction rate

Off-policy operators with unbiased fixed point $\mathbb{B}(\mathcal{T}) = 0$ are usually more conducive to theoretical analysis [3, 7]. For example, Retrace operators $\mathcal{R}_c^{\pi, \mu}$ are a family of off-policy evaluation operators indexed by trace coefficients $c(x, a)$. When $c(x, a) \leq \pi(a | x) / \mu(a | x)$, these operators are unbiased in that $\mathcal{R}_c^{\pi, \mu} Q^\pi = Q^\pi$, resulting in $\mathbb{B}(\mathcal{R}_c^{\pi, \mu}) = 0$. One popular choice is $c(x, a) = \min\{\bar{c}, \pi(a | x) / \mu(a | x)\}$ such that the operator also controls variance $\mathbb{V}(\mathcal{R}_c^{\pi, \mu})$ [3] with \bar{c} .

However, many prior empirical results suggest that bias is not a major bottleneck in practice. For example, uncorrected n -step update is a popular technique which greatly improves DQN [14] where the RL agent applies the operator $\mathcal{T}_{\text{nstep}}^{\pi, \mu} := (\mathcal{T}^\mu)^{n-1} \mathcal{T}^\pi$ where π, μ are target and behavior policies respectively [5, 6]. Note that since $\mathcal{T}_{\text{nstep}}^{\pi, \mu} Q^\pi \neq Q^\pi$, the n -step operator is biased $\mathbb{B}(\mathcal{T}_{\text{nstep}}^{\pi, \mu}) > 0$ [7]. However, its contraction rate is small due to uncorrected updates $\Gamma(\mathcal{T}_{\text{nstep}}^{\pi, \mu}) \leq \gamma^n$. On the other hand, though Retrace operators have unbiased fixed point, its contraction rates are typically high due to small IS, which *cut off* the signals early and fail to bootstrap with long horizons. The relative importance of contraction rate over bias is confirmed through the empirical observations that n -step often performs significantly better than Retrace in challenging domains [6, 7]. Such observations also motivate trading off bias and contraction rates in an adaptive way [7].

2.3 Self-imitation Learning

Maximum entropy RL. SIL is established under the framework of maximum-entropy RL [19–23], where the reward is augmented by an entropy term $r_{\text{ent}}(x, a) := r(x, a) + c\mathcal{H}^\pi(x)$ and $\mathcal{H}^\pi(x)$ is the entropy of policy π at state x , weighted by a constant $c > 0$. Accordingly, the Q-function is $Q_{\text{ent}}^\pi(x_0, a_0) := \mathbb{E}_\pi[r_0 + \sum_{t \geq 1} \gamma^t (r_t + c\mathcal{H}^\pi(x_t))]$. The maximum-entropy RL objective is $J_{\text{ent}}(\pi) := \mathbb{E}_\pi[\sum_{t \geq 0} \gamma^t (r_t + c\mathcal{H}^\pi(x_t))]$. Similar to standard RL, we denote the optimal policy $\pi_{\text{ent}}^* = \arg \max_\pi J_{\text{ent}}(\pi)$ and its Q-function $Q_{\text{ent}}^{\pi_{\text{ent}}^*}(x, a)$.

Lower bound Q-learning. Lower bound Q-learning is motivated by the following inequality [8],

$$Q_{\text{ent}}^{\pi_{\text{ent}}^*}(x, a) \geq Q_{\text{ent}}^\mu(x, a) = \mathbb{E}_\mu[r_0 + \sum_{t \geq 1} \gamma^t (r_t + c\mathcal{H}^\mu(x_t))], \quad (2)$$

where μ is an arbitrary behavior policy. Lower bound Q-learning optimizes the following objective with the parameterized Q-function $Q_\theta(x, a)$,

$$\min_{\theta} \mathbb{E}_{\mathcal{D}}[(Q^\mu(x, a) - Q_\theta(x, a))_+]^2, \quad (3)$$

where $[x]_+ := \max(x, 0)$. The intuition of Eqn.(3) is that the Q-function $Q_\theta(x, a)$ obtains learning signals from all trajectories such that $Q_{\text{ent}}^\mu(x, a) > Q_\theta(x, a) \approx Q^{\pi_\theta}(x, a)$, i.e. trajectories which perform better than the current policy π_θ . In practice $Q_{\text{ent}}^\mu(x, a)$ could be estimated via a single trajectory $Q_{\text{ent}}^\mu(x, a) \approx \hat{R}^\mu(x, a) := r_0 + \sum_{t \geq 1} \gamma^t (r_t + c\mathcal{H}^\mu(x_t))$. Though in Eqn.(3) one could plug in $\hat{R}^\mu(x, a)$ in place of $Q^\mu(x, a)$ [8, 10], this introduces bias due to the double-sample issue [24], especially when $\hat{R}^\mu(x, a)$ has high variance either due to the dynamics or a stochastic policy.

SIL with stochastic actor-critic. SIL further focuses on actor-critic algorithms where the Q-function is parameterized by a value-function and a stochastic policy $Q_\theta(x, a) := V_\theta(x) + c \log \pi_\theta(a | x)$. Taking gradients of the loss in Eqn.(3) with respect to θ yields the following loss function of the value-function and policy. The full SIL loss is $L_{\text{sil}}(\theta) = L_{\text{value}}(\theta) + L_{\text{policy}}(\theta)$.

$$L_{\text{value}}(\theta) = \frac{1}{2}([\hat{R}^\mu(x, a) - V_\theta(x)]_+)^2, L_{\text{policy}}(\theta) = -\log \pi_\theta(a | x)[\hat{R}^\mu(x, a) - V_\theta(x)]_+. \quad (4)$$

3 Generalized Self-Imitation Learning

3.1 Generalized Lower Bounds for Optimal Q-functions

To generalize the formulation of SIL, we seek to provide generalized lower bounds for the optimal Q-functions. Practical lower bounds should possess several desiderata: **(P.1)** they could be estimated using off-policy partial trajectories; **(P.2)** they could bootstrap from learned Q-functions.

In standard actor-critic algorithms, partial trajectories are generated via behavior policy μ (for example, see [25, 18, 26]), and the algorithm maintains an estimate of Q-functions for the current policy π . The following theorem states a general lower bound for the max-entropy optimal Q-function $Q_{\text{ent}}^{\pi^*}$. Additional results on generalized lower bounds of the optimal value function V^{π^*} could be similarly derived, and we leave its details in Theorem 3 in Appendix C.

Theorem 1. (proof in Appendix A) *Let π_{ent}^* be the optimal policy and $Q_{\text{ent}}^{\pi_{\text{ent}}^*}$ its Q-function under maximum entropy RL formulation. Given a partial trajectory $(x_t, a_t)_{t=0}^n$, the following inequality holds for any n ,*

$$Q_{\text{ent}}^{\pi_{\text{ent}}^*}(x_0, a_0) \geq L_{\text{ent}}^{\pi, \mu, n}(x_0, a_0) := \mathbb{E}_{\mu}[r_0 + \gamma c \mathcal{H}^{\mu}(x_1) + \sum_{t=1}^{n-1} \gamma^t (r_t + c \mathcal{H}^{\mu}(x_{t+1})) + \gamma^n Q_{\text{ent}}^{\pi}(x_n, a_n)] \quad (5)$$

By letting $c = 0$, we derive a generalized lower bound for the standard optimal Q-function Q^{π^*}

Lemma 1. *Let π^* be the optimal policy and Q^{π^*} its Q-function under standard RL. Given a partial trajectory $(x_t, a_t)_{t=0}^n$, the following inequality holds for any n ,*

$$Q^{\pi^*}(x_0, a_0) \geq L^{\pi, \mu, n}(x_0, a_0) := \mathbb{E}_{\mu}[\sum_{t=0}^{n-1} \gamma^t r_t + \gamma^n Q^{\pi}(x_n, a_n)]. \quad (6)$$

We see that n -step lower bounds $L_{\text{ent}}^{\pi, \mu, n}$ satisfy both desiderata **(P.1)(P.2)**: $L_{\text{ent}}^{\pi, \mu, n}$ could be estimated on a single trajectory and bootstraps from learned Q-function $Q_{\theta}(x, a) \approx Q^{\pi}(x, a)$. When $n \rightarrow \infty$, $L_{\text{ent}}^{\pi, \mu, n} \rightarrow Q^{\mu}$ and we arrive at the lower bound employed by the original SIL [8]. The original SIL does not satisfy **(P.1)(P.2)**: the estimate of Q^{μ} requires full trajectories from finished episodes and does not bootstrap from learned Q-functions. In addition, because the lower bound $L_{\text{ent}}^{\pi, \mu}(x, a)$ bootstraps Q-functions at a finite step n , we expect it to partially mitigate the double-sample bias of $\hat{R}^{\mu}(x, a)$. Also, as the policy π improves over time, the Q-function $Q^{\pi}(x, a)$ increases and the bound $L^{\pi, \mu, n}$ improves as well. On the contrary, the standard SIL does not enjoy such advantages.

3.2 Generalized Self-Imitation Learning

Generalized SIL with stochastic actor-critic. We describe the generalized SIL for actor-critic algorithms. As developed in Section 2.3, such algorithms maintain a parameterized *stochastic* policy $\pi_{\theta}(a | x)$ and value-function $V_{\theta}(x)$. Let $\hat{L}_{\text{ent}}^{\pi, \mu, n}(x, a)$ denote the sample estimate of the n -step lower bound, the loss functions are

$$L_{\text{value}}^{(n)}(\theta) = \frac{1}{2}([\hat{L}_{\text{ent}}^{\pi, \mu, n}(x, a) - V_{\theta}(x)]_+)^2, L_{\text{policy}}^{(n)}(\theta) = -\log \pi_{\theta}(a | x)[\hat{L}_{\text{ent}}^{\pi, \mu, n}(x, a) - V_{\theta}(x)]_+. \quad (7)$$

Note that the loss functions in Eqn.(7) introduce updates very similar to A2C [25]. Indeed, when removing the threshold function $[x]_+$ and setting the data distribution to be on-policy $\mu = \pi$, we recover the n -step A2C objective.

Generalized SIL with deterministic actor-critic. For continuous control, temporal difference (TD)-learning and deterministic policy gradients have proven highly sample efficient and high-performing [15, 27, 23]. By construction, the generalized n -step lower bounds $L_{\text{ent}}^{\pi, \mu, n}$ adopts n -step TD-learning and should naturally benefit the aforementioned algorithms. Such algorithms maintain a parameterized Q-function $Q_{\theta}(x, a)$, which could be directly updated via the following loss

$$L_{\text{qvalue}}^{(n)}(\theta) = \frac{1}{2}([\hat{L}_{\text{ent}}^{\pi, \mu, n}(x, a) - Q_{\theta}(x, a)]_+)^2. \quad (8)$$

Interestingly, note that the above update Eqn.(8) is similar to n -step Q-learning update [4, 5] up to the threshold function $[x]_+$. In Section 4, we will discuss their formal connections in details.

Prioritized experience replay. Prior work on prioritized experience replay [28, 29] proposed to sample tuples (x_t, a_t, r_t) from replay buffer \mathcal{D} with probability proportional to Bellman errors. We provide a straightforward extension by sampling proportional to the lower bound loss $[\hat{L}_{\text{ent}}^{\pi, \mu, n}(x, a) - Q_\theta(x, a)]_+$. This reduces to the sampling scheme in SIL [8] when letting $n \rightarrow \infty$.

4 Trade-offs with Lower Bound Q-learning

When applying SIL in practice, its induced loss functions are optimized jointly with the base loss functions [8]: in the case of stochastic actor-critic, the full loss function is $L(\theta) := L_{\text{ac}}(\theta) + L_{\text{sil}}(\theta)$, where $L_{\text{ac}}(\theta)$ is the original actor-critic loss function [25]. The parameter is then updated via the gradient descent step $\theta = \theta - \nabla_\theta L(\theta)$. This makes it difficult to analyze the behavior of SIL beyond the plain motivation of Q-function lower bounds. Though a comprehensive analysis of SIL might be elusive due to its empirical nature, we formalize the lower bound arguments via RL operators and draw connections with n -step Q-learning. Below, we present results for standard RL.

4.1 Operators for Generalized Lower Bound Q-learning

First, we formalize the mathematical operator of SIL. Let $Q \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{A}|}$ be a vector-valued Q-function. Given some behavior policy μ , define the operator $\mathcal{T}_{\text{sil}}Q(x, a) := Q(x, a) + [Q^\mu(x, a) - Q(x, a)]_+$ where $[x]_+ := \max(x, 0)$. This operator captures the defining feature of the practical lower bound Q-learning [8], where the Q-function $Q(x, a)$ receives learning signals only when $Q^\mu(x, a) > Q(x, a)$. For generalized SIL, we similarly define $\mathcal{T}_{n, \text{sil}}Q(x, a) := Q(x, a) + [(\mathcal{T}^\mu)^{n-1}\mathcal{T}^\pi Q(x, a) - Q(x, a)]_+$, where $Q(x, a)$ is updated when $(\mathcal{T}^\mu)^{n-1}\mathcal{T}^\pi Q(x, a) > Q(x, a)$ as suggested in Eqn.(7,8).

In practice, lower bound Q-learning is applied alongside other main iterative algorithms. Henceforth, we focus on policy iteration algorithms with the Bellman operator \mathcal{T}^π along with its n -step variant $(\mathcal{T}^\mu)^{n-1}\mathcal{T}^\pi$. Though practical deep RL implementations adopt additive loss functions, for theoretical analysis we consider a convex combination of these three operators, with coefficients $\alpha, \beta \in [0, 1]$.

$$\mathcal{T}_{n, \text{sil}}^{\alpha, \beta} := (1 - \beta)\mathcal{T}^\pi + (1 - \alpha)\beta\mathcal{T}_{n, \text{sil}} + \alpha\beta(\mathcal{T}^\mu)^{n-1}\mathcal{T}^\pi \quad (9)$$

4.2 Properties of the operators

Theorem 2. (proof in Appendix B) *Let π, μ be target and behavior policy respectively. Then the following results hold:*

- **Contraction rate.** $\Gamma(\mathcal{T}_{n, \text{sil}}^{\alpha, \beta}) \leq (1 - \beta)\gamma + (1 - \alpha)\beta + \alpha\beta\gamma^n$. The operator is always contractive for $\alpha \in [0, 1], \beta \in [0, 1]$. When $\alpha > \frac{1-\gamma}{1-\gamma^n}$, we have for any $\beta \in (0, 1)$, $\Gamma(\mathcal{T}_{n, \text{sil}}^{\alpha, \beta}) \leq \gamma' < \gamma$ for some γ' .
- **Fixed point bias.** $\mathcal{T}_{n, \text{sil}}^{\alpha, \beta}$ has a unique fixed point $\tilde{Q}^{\alpha, \beta}$ for any $\alpha \in [0, 1], \beta \in [0, 1]$ such that $(1 - \alpha)\beta < 1$. This fixed point satisfies the bounds $Q^{\eta\pi + (1-\eta)\mu^{n-1}\pi} \leq \tilde{Q}^{\alpha, \beta} \leq Q^{\pi^*}$, where $Q^{\eta\pi + (1-\eta)\mu^{n-1}\pi}$ is the unique fixed point of operator $\eta\mathcal{T}^\pi + (1 - \eta)\mathcal{T}^{\mu^{n-1}\pi}$ with $\eta = \frac{1-\beta}{1-\beta+\alpha\beta}$.

To highlight the connections between uncorrected n -step and SIL, we discuss two special cases.

- When $\alpha = 1$, $\mathcal{T}_{n, \text{sil}}^{\alpha, \beta}$ removes all the lower bound components and reduces to $(1 - \beta)\mathcal{T}^\pi + \beta(\mathcal{T}^\mu)^{n-1}\mathcal{T}^\pi$. This recovers the trade-off results discussed in [7]: when $\beta = 1$, the operator becomes uncorrected n -step updates with the smallest possible contraction rate $\Gamma(\mathcal{T}_{n, \text{sil}}^{\alpha, \beta}) \leq \gamma^n$, but the fixed point $\tilde{Q}^{\alpha, \beta}$ is biased. In general, there is no lower bound on the fixed point so that its value could be arbitrary depending on both π and μ .
- When $\alpha \in (\frac{1-\gamma}{1-\gamma^n}, 1]$, $\mathcal{T}_{n, \text{sil}}^{\alpha, \beta}$ combines the lower bound operator. Importantly, unlike uncorrected n -step, now the fixed point is lower bounded $\tilde{Q}^{\alpha, \beta} \geq Q^{\eta\pi + (1-\eta)\mu^{n-1}\pi}$. Because such a fixed point bias is lower bounded, we call it *positive bias*. Adjusting α creates a trade-off between contraction rates and the positive fixed point bias. In addition, the fixed point bias is safe in that it is upper

bounded by the optimal Q-function, $\tilde{Q}^{\alpha,\beta} \leq Q^{\pi^*}$, which might be a desirable property in cases where over-estimation bias hurts the practical performance [30, 27]. In Section 5, we will see that such positive fixed point bias is beneficial to empirical performance, as similarly observed in [11, 8, 10]. Though $\mathcal{T}_{n,\text{sil}}^{\alpha,\beta}$ does not contract as fast as the uncorrected n -step operator $(\mathcal{T}^\mu)^{n-1}\mathcal{T}^\pi$, it still achieves a bound on contraction rates strictly smaller than \mathcal{T}^π . As such, generalized SIL also enjoys fast contractions relative to the baseline algorithm.

Empirical evaluation of Q-function bias. To validate the statements made in Theorem 2 on the bias of Q-functions, we test with TD3 for an empirical evaluation [27]. At a given time in training, the bias at a pair (x, a) is calculated as the difference between Q-function network prediction and an unbiased Monte-Carlo estimate of Q-function for the current policy π , i.e. $Q_\theta(x, a) - \hat{Q}^\pi(x, a)$ ¹. Figure 1 shows the mean $\pm 0.5\text{std}$ of such bias over time, with mean and std computed over visited state-action pairs under π . In general, the bias of TD3 is small, which is compatible to observations made in [27]. The bias of TD3 with uncorrected $n = 5$ -step spreads over a wider range near zero, indicating significant non-zero bias on both sides. For TD3 with generalized SIL $n = 5$, the bias is also spread out but the mean bias is significantly greater than zero. This implies that SIL generally induces a positive bias in the fixed point. In summary, these observations confirm that neural network based Q-functions $Q_\theta(x, a)$ display similar biases introduced by the corresponding exact operators.

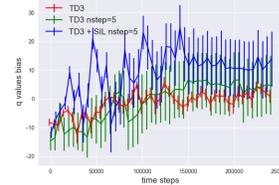


Figure 1: Bias of Q-function networks with twin-delayed deep deterministic policy gradient (TD3) variants on the WalkerStand task.

5 Experiments

We seek to address the following questions in the experiments: **(1)** Does generalized SIL entail performance gains on both deterministic and stochastic actor-critic algorithms? **(2)** How do the design choices (e.g. hyper-parameters, prioritized replay) of generalized SIL impact its performance?

Benchmark tasks. For benchmark tasks, we focus on state-based continuous control. In order to assess the strengths of different algorithmic variants, we consider similar tasks *Walker*, *Cheetah* and *Ant* with different simulation backends from OpenAI gym [31], DeepMind Control Suite [32] and Bullet Physics Engine [33]. These backends differ in many aspects, e.g. dimensions of observation and action space, transition dynamics and reward functions. With such a wide range of varieties, we seek to validate algorithmic gains with sufficient robustness to varying domains. There are a total of 8 distinct simulated control tasks, with details in Appendix D.

5.1 Deterministic actor-critic

Baselines. We choose TD3 [27] as the baseline algorithm which employs a deterministic actor $\pi_\phi(x)$. TD3 builds on deep deterministic policy gradient (DDPG) [15] and alleviates the over-estimation bias in DDPG via delayed updates and double critics similar to double Q-learning [34, 30]. Through a comparison of DDPG and TD3 combined with generalized SIL, we will see that over-estimation bias makes the advantages through lower bound Q-learning much less significant. To incorporate generalized SIL into TD3, we adopt an additive loss function: let $L_{\text{TD3}^{(n)}}(\theta)$ be the n -step TD3 loss function and $L_{\text{sil}}^{(m)}(\theta)$ be the m -step generalized SIL loss. The full loss is $L(\theta) := L_{\text{TD3}^{(n)}}(\theta) + \eta L_{\text{sil}}^{(m)}(\theta)$ with some $\eta \geq 0$. We will use this general loss template to describe algorithmic variants for comparison below.

Return-based SIL for TD3. A straightforward extension of SIL [8] and optimality tightening [11] to deterministic actor-critic algorithms, is to estimate the return $\hat{R}^\mu(x_t, a_t) := \sum_{t' \geq t} \gamma^{t-t'} r_{t'}$ on a

¹By definition, the bias should be the difference between the fixed point \tilde{Q} and target Q^π . Since TD3 employs heavy replay during training, we expect the Q-function to be close to the fixed point $Q_\theta \approx \tilde{Q}$. Because both the dynamics and policy are deterministic, an one-sample estimate of Q-function is accurate enough to approximate the true Q-function $\hat{Q}^\pi = Q^\pi$. Hence here the bias is approximated by $Q_\theta - \hat{Q}^\pi$.

single trajectory $(x_t, a_t, r_t)_{t=0}^{\infty}$ and minimize the lower bound objective $([\hat{R}^{\mu}(x, a) - Q_{\theta}(x, a)]_+)^2$. Note that since both the policy and the transition is deterministic (for benchmarks listed above), the one-sample estimate of returns is exact in that $\hat{R}^{\mu}(x, a) \equiv R^{\mu}(x, a) \equiv Q^{\mu}(x, a)$. In this case, return-based SIL is exactly equivalent to generalized SIL with $n \rightarrow \infty$.

Evaluations. We provide evaluations on a few standard benchmark tasks in Figure 2 as well as their variants with delayed rewards. To facilitate the credit assignment of the training performance to various components of the generalized SIL, we compare with a few algorithmic variants: 1-step TD3 ($n = 1, \eta = 0$); 5-step TD3 ($n = 5, \eta = 0$); TD3 with 5-step generalized SIL ($n = 1, \eta = 0.1, m = 5$); TD3 with return-based SIL ($n = 1, \eta = 0.1, m = \infty$). Importantly, note that the weighting coefficient is fixed $\eta = 0.1$ for all cases of generalized SIL. The training results of selected algorithms are shown in Figure 2. We show the final performance of all baselines in Table 1 in Appendix D.

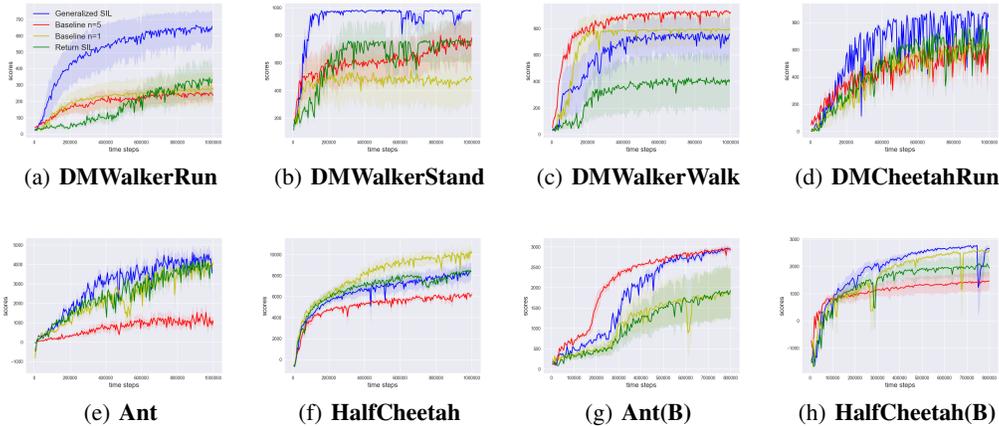


Figure 2: Standard evaluations on 8 benchmark tasks. Different colors represent different algorithmic variants. Each curve shows the mean ± 0.5 std of evaluation performance during training, averaged across 3 random seeds. The x-axis shows the time steps and the y-axis shows the cumulative returns. Observe that 5-step generalized SIL (blue) generally outperforms other baselines. Tasks with *DM* are from DeepMind Control Suite, and tasks with *(B)* are from Bullet.

We make several observations: (1) For uncorrected n -step, the best n is task dependent. However, 5-step generalized SIL consistently improves the performance over uncorrected n -step TD3 baselines; (2) SIL losses generally accelerate the optimization. Indeed, both generalized SIL and return-based SIL generally performs better than pure TD3 algorithms; (3) The advantage of generalized SIL is more than n -step bootstrap. Because n -step generalized SIL is similar to n -step updates, it is reasonable to speculate that the performance gains of SIL are partly attributed to n -step updates. By the significant advantages of generalized SIL relative to n -step updates, we see that its performance gains also come from the lower bound techniques; (4) n -step SIL with $n = 5$ works the best. With $n = 1$, SIL does not benefit from bootstrapping partial trajectories with long horizons; with $n = \infty$, SIL does not benefit from bootstrapped values at all. As discussed in Section 3, n -step bootstrap provides benefits in (i) variance reduction (replacing the discounted sum of rewards by a value function) and (ii) tightened bounds. In deterministic environment with deterministic policy, the advantage (ii) leads to most of the performance gains.

5.2 Ablation study for deterministic actor-critic

Please refer to Table 1 in Appendix D for a summary of ablation experiments over SIL variants. Here, we focus on discussions of the ablation results.

Horizon parameter n . In our experience, we find that $n = 5$ works reasonably well though other close values might work as well. To clarify the extreme effect of n : at one extreme, $n = 1$ and SIL does not benefit from trajectory-based learning and generally underperforms $n = 5$; when $n = \infty$, the return-based SIL does not provide as significant speed up as $n = 5$.

Prioritized experience replay. In general, prioritized replay has two hyper-parameters: α for the degree of prioritized sampling and β for the degree of corrections [28]. For general SIL, we adopt $\alpha = 0.6, \beta = 0.1$ as in [8]. We also consider variants where the tuples are sampled according to the priority but IS weights are not corrected ($\alpha = 0.6, \beta = 0.0$) and where there is no prioritized sampling ($\alpha = \beta = 0.0$). The results are reported in Table 2 in Appendix D. We observe that generalized SIL works the best when both prioritized sampling and IS corrections are present.

Over-estimation bias. Algorithms with over-estimation bias (e.g. DDPG) does not benefit as much (e.g. TD3) from the lower bound loss, as shown by additional results in Appendix D. We speculate that this is because by construction the Q-function network $Q_\theta(x, a)$ should be a close approximation to the Q-function $Q^\pi(x, a)$. In cases where over-estimation bias is severe, this assumption does not hold. As a result, the performance is potentially harmed instead of improved by the *uncontrolled* positive bias [30, 27]. This contrasts with the *controlled* positive bias of SIL, which improves the performance.

5.3 Stochastic actor-critic

Baselines. For the stochastic actor-critic algorithm, we adopt proximal policy optimization (PPO) [18]. Unlike critic-based algorithms such as TD3, PPO estimates gradients using near on-policy samples.

Delayed reward environments. Delayed reward environment tests algorithms’ capability to tackle delayed feedback in the form of sparse rewards [8]. In particular, a standard benchmark environment returns dense reward r_t at each step t . Consider accumulating the reward over d consecutive steps and return the sum at the end k steps, i.e. $r'_t = 0$ if $t \bmod k \neq 0$ and $r'_t = \sum_{\tau=t-d+1}^t r_\tau$ if $t \bmod d = 0$.

Evaluations. We compare three baselines: PPO, PPO with SIL [8] and PPO with generalized SIL with $n = 5$ -step. We train these variants on a set of OpenAI gym tasks with delayed rewards, where the delays are $k \in \{1, 5, 10, 20\}$. Please refer to Appendix D for further details of the algorithms. The final performance of algorithms after training ($5 \cdot 10^6$ steps for HalfCheetah and 10^7 for the others) are shown in Figure 3. We make several observations: **(1)** The performance of PPO is generally inferior to its generalized SIL or SIL extensions. This implies the necessity of carrying out SIL in general, as observed in [8]; **(2)** The performance of generalized SIL with $n = 5$ differ depending on the tasks. SIL works significantly better with Ant, while generalized SIL works better with Humanoid. Since SIL is a special case for $n \rightarrow \infty$, this implies the potential benefits of adapting n for each task.

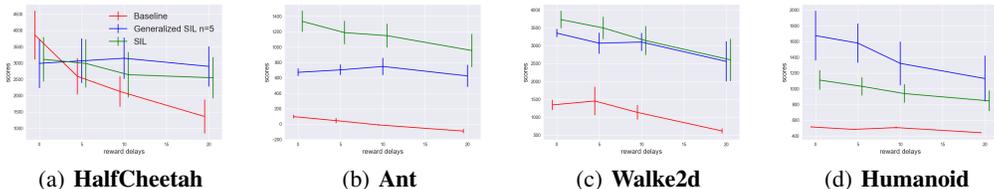


Figure 3: Standard evaluations on 4 benchmark OpenAI gym tasks. Different colors represent different algorithmic variants. Each curve shows the mean ± 0.5 std of evaluation performance at the end of training, averaged across 5 random seeds. The x-axis shows the delayed time steps for rewards and the y-axis shows the cumulative returns. The ticks $\{1, 5, 10, 20\}$ show the delays and the x-axis of the plotted data is slightly shifted for better visualization.

6 Further Discussions on Related Work

Over-estimation bias in Q-learning. Q-learning and TD-learning are popular algorithms for RL [35, 36]. Due to the max operator, sampled updates of Q-learning naturally incur over-estimation bias, which potentially leads to unstable learning. To mitigate the bias, prior work has considered Double Q-learning [34, 30], explicit bias correction [37], linear combination between Double Q-learning and Q-learning [38], bootstrapping from past predictions [39] and using an ensemble of Q-functions

[40]. Similar ideas have been applied to actor-critic algorithms [27]. While it is conventionally believed that over-estimation bias is hurtful to the performance, [40] provides concrete examples where estimation bias (under- or over-estimation) could accelerate learning. In practice, for certain RL environments where rewards are sparse, it is desirable to introduce positive bias to encourage exploration [8].

Learning from off-policy data. Off-policy learning is crucial for modern RL algorithms [41, 42]. At the core of many off-policy learning algorithms [1, 43, 44, 3, 45, 46], importance sampling (IS) corrects for the distributional mismatch between behavior π and target policy μ , generating *unbiased* updates. Despite the theoretical foundations, IS-based algorithms often underperform empirically motivated algorithms such as n -step updates [4–6]. In general, uncorrected n -step algorithms could be interpreted as trading-off fast contractions with fixed point bias [7], which seems to have a significant practical effect. In addition to potentially better performance, uncorrected n -step updates also do not require e.g. $\mu(a | x)$. This entails learning with truly arbitrary off-policy data. Built on top of n -step updates, we propose generalized n -step SIL which intentionally introduces a positive bias into the fixed point, effectively filtering out behavior data with poor performance. This idea of learning from good-performing off-policy data is rooted in algorithmic paradigms such as behavior cloning [47], inverse RL [48], and more recently instantiated by e.g., episodic control [49, 50] lower bound Q-learning [11] and SIL [8–10].

7 Conclusion

We have proposed generalized n -step lower bound Q-learning, a strict generalization of return-based lower bound Q-learning and the corresponding self-imitation learning algorithm [8]. We have drawn close connections between n -step lower bound Q-learning and uncorrected n -step updates: both techniques achieve performance gains by invoking a trade-off between contraction rates and fixed point bias of the evaluation operators. Empirically, we observe that the positive bias induced by lower bound Q-learning provides more consistent improvements than arbitrary n -step bias. It is of interest to study in general what bias could be beneficial to policy optimization, and how to exploit such bias in practical RL algorithms.

8 Broader Impact

Algorithms which learn from off-policy samples are critical for the applications of RL to more impactful real life domains such as autonomous driving and health care. Our work provides insights into SIL, and its close connections to popular off-policy learning techniques such as n -step Q-learning. We believe our work entails a positive step towards better understanding of efficient off-policy RL algorithms, which paves the way for future research into important applications.

9 Acknowledgements

The author thanks Mark Rowland and Tadashi Kozuno for insightful discussions about this project.

References

- [1] Doina Precup, Richard S Sutton, and Sanjoy Dasgupta. Off-policy temporal-difference learning with function approximation. In *ICML*, pages 417–424, 2001.
- [2] Anna Harutyunyan, Marc G Bellemare, Tom Stepleton, and Rémi Munos. Q (λ) with off-policy corrections. In *International Conference on Algorithmic Learning Theory*, pages 305–320. Springer, 2016.
- [3] Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc Bellemare. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1054–1062, 2016.
- [4] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [5] 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.
- [6] Steven Kapturowski, Georg Ostrovski, John Quan, Remi Munos, and Will Dabney. Recurrent experience replay in distributed reinforcement learning. 2018.
- [7] Mark Rowland, Will Dabney, and Rémi Munos. Adaptive trade-offs in off-policy learning. *arXiv preprint arXiv:1910.07478*, 2019.
- [8] Junhyuk Oh, Yijie Guo, Satinder Singh, and Honglak Lee. Self-imitation learning. *arXiv preprint arXiv:1806.05635*, 2018.
- [9] Tanmay Gangwani, Qiang Liu, and Jian Peng. Learning self-imitating diverse policies. *arXiv preprint arXiv:1805.10309*, 2018.
- [10] Yijie Guo, Jongwook Choi, Marcin Moczulski, Samy Bengio, Mohammad Norouzi, and Honglak Lee. Efficient exploration with self-imitation learning via trajectory-conditioned policy. *arXiv preprint arXiv:1907.10247*, 2019.
- [11] Frank S He, Yang Liu, Alexander G Schwing, and Jian Peng. Learning to play in a day: Faster deep reinforcement learning by optimality tightening. *arXiv preprint arXiv:1611.01606*, 2016.
- [12] Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, pages 679–684, 1957.
- [13] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.
- [14] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.
- [15] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.
- [16] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 2015.
- [17] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando De Freitas. Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581*, 2015.

- [18] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [19] Brian D Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University, 2010.
- [20] Roy Fox, Ari Pakman, and Naftali Tishby. Taming the noise in reinforcement learning via soft updates. *arXiv preprint arXiv:1512.08562*, 2015.
- [21] Kavosh Asadi and Michael L Littman. An alternative softmax operator for reinforcement learning. In *International Conference on Machine Learning*, pages 243–252, 2017.
- [22] Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Bridging the gap between value and policy based reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2775–2785, 2017.
- [23] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.
- [24] Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pages 30–37. Elsevier, 1995.
- [25] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937, 2016.
- [26] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Volodymyr Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. *arXiv preprint arXiv:1802.01561*, 2018.
- [27] Scott Fujimoto, Herke Van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*, 2018.
- [28] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.
- [29] 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.
- [30] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI conference on artificial intelligence*, 2016.
- [31] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.
- [32] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.
- [33] Erwin Coumans. Bullet physics engine. *Open Source Software: <http://bulletphysics.org>*, 1(3):84, 2010.
- [34] Hado V Hasselt. Double q-learning. In *Advances in neural information processing systems*, pages 2613–2621, 2010.
- [35] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.
- [36] Dimitri P Bertsekas and John N Tsitsiklis. Neuro-dynamic programming: an overview. In *Proceedings of 1995 34th IEEE Conference on Decision and Control*, volume 1, pages 560–564. IEEE, 1995.

- [37] Donghun Lee, Boris Defourny, and Warren B Powell. Bias-corrected q-learning to control max-operator bias in q-learning. In *2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 93–99. IEEE, 2013.
- [38] Zongzhang Zhang, Zhiyuan Pan, and Mykel J Kochenderfer. Weighted double q-learning. In *IJCAI*, pages 3455–3461, 2017.
- [39] Oron Anschel, Nir Baram, and Nahum Shimkin. Averaged-dqn: Variance reduction and stabilization for deep reinforcement learning. In *International Conference on Machine Learning*, pages 176–185. PMLR, 2017.
- [40] Qingfeng Lan, Yangchen Pan, Alona Fyshe, and Martha White. Maxmin q-learning: Controlling the estimation bias of q-learning. *arXiv preprint arXiv:2002.06487*, 2020.
- [41] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.
- [42] Csaba Szepesvári. Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning*, 4(1):1–103, 2010.
- [43] Miroslav Dudík, Dumitru Erhan, John Langford, Lihong Li, et al. Doubly robust policy evaluation and optimization. *Statistical Science*, 29(4):485–511, 2014.
- [44] Philip Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 2139–2148, 2016.
- [45] Ashique Rupam Mahmood, Huizhen Yu, and Richard S Sutton. Multi-step off-policy learning without importance sampling ratios. *arXiv preprint arXiv:1702.03006*, 2017.
- [46] Mehrdad Farajtabar, Yinlam Chow, and Mohammad Ghavamzadeh. More robust doubly robust off-policy evaluation. *arXiv preprint arXiv:1802.03493*, 2018.
- [47] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635, 2011.
- [48] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004.
- [49] Charles Blundell, Benigno Uria, Alexander Pritzel, Yazhe Li, Avraham Ruderman, Joel Z Leibo, Jack Rae, Daan Wierstra, and Demis Hassabis. Model-free episodic control. *arXiv preprint arXiv:1606.04460*, 2016.
- [50] Alexander Pritzel, Benigno Uria, Sriram Srinivasan, Adria Puigdomenech, Oriol Vinyals, Demis Hassabis, Daan Wierstra, and Charles Blundell. Neural episodic control. *arXiv preprint arXiv:1703.01988*, 2017.
- [51] Joshua Achiam. Openai spinning up. *GitHub, GitHub repository*, 2018.
- [52] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [53] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.
- [54] Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. Openai baselines. <https://github.com/openai/baselines>, 2017.

A Proof of Theorem 1

Recall that under maximum entropy RL, the Q-function is defined as $Q_{\text{ent}}^\pi(x_0, a_0) := \mathbb{E}_\pi[r_0 + \sum_{t=1}^{\infty} \gamma^t (r_t + c\mathcal{H}^\pi(x_t))]$ where $\mathcal{H}^\mu(x_t)$ is the entropy of the distribution $\pi^\mu(\cdot | x_t)$. The Bellman equation for Q-function is naturally

$$Q^\pi(x_0, a_0) = \mathbb{E}_\pi[r_0 + \gamma c\mathcal{H}^\pi(x_1) + \gamma Q^\pi(x_1, a_1)].$$

Let the optimal policy be π_{ent}^* . The relationship between the optimal policy and its Q-function is $\pi_{\text{ent}}^*(a | x) \propto \exp(Q_{\text{ent}}^{\pi_{\text{ent}}^*}(x, a)/c)$. We seek to establish $Q_{\text{ent}}^{\pi_{\text{ent}}^*}(x_0, a_0) \geq \mathbb{E}_\mu[r_0 + \gamma c\mathcal{H}^\mu(x_1) + \sum_{t=1}^{T-1} \gamma^t (r_t + c\mathcal{H}^\mu(x_{t+1})) + \gamma^T Q_{\text{ent}}^\pi(x_T, a_T)]$ for any policy μ, π .

We prove the results using induction. For the base case $T = 1$,

$$\begin{aligned} Q_{\text{ent}}^{\pi_{\text{ent}}^*}(x_0, a_0) &= \mathbb{E}_{\pi_{\text{ent}}^*}[r_0 + \gamma c\mathcal{H}^{\pi_{\text{ent}}^*}(x_1) + \gamma Q_{\text{ent}}^{\pi_{\text{ent}}^*}(x_1, a_1)] \\ &= \mathbb{E}_{x_1 \sim p(\cdot | x_0, a_0)}[r_0 + \gamma c\mathcal{H}^{\pi_{\text{ent}}^*}(x_1) + \gamma \mathbb{E}_{\pi_{\text{ent}}^*}[Q_{\text{ent}}^{\pi_{\text{ent}}^*}(x_1, a_1)]] \\ &\geq \mathbb{E}_{x_1 \sim p(\cdot | x_0, a_0)}[r_0 + \gamma c\mathcal{H}^\mu(x_1) + \gamma \mathbb{E}_\mu[Q_{\text{ent}}^{\pi_{\text{ent}}^*}(x_1, a_1)]] \\ &\geq \mathbb{E}_{x_1 \sim p(\cdot | x_0, a_0)}[r_0 + \gamma c\mathcal{H}^\mu(x_1) + \gamma \mathbb{E}_\mu[Q_{\text{ent}}^\pi(x_1, a_1)]]. \end{aligned} \tag{10}$$

In the above, to make the derivations clear, we single out the reward r_0 and state $x_1 \sim p(\cdot | x_0, a_0)$, note that the distributions of these two quantities do not depend on the policy. The first inequality follows from the fact that $\pi_{\text{ent}}^*(\cdot | x) = \arg \max_\pi [c\mathcal{H}^\pi(x) + \mathbb{E}_{a \sim \pi(\cdot | x)} Q_{\text{ent}}^{\pi_{\text{ent}}^*}(x, a)]$. The second inequality follows from $Q_{\text{ent}}^{\pi_{\text{ent}}^*}(x, a) \geq Q_{\text{ent}}^\pi(x, a)$ for any policy π .

With the base case in place, assume that the result holds for $T \leq k - 1$. Consider the case $T = k$

$$\begin{aligned} \mathbb{E}_\mu[r_0 + \gamma c\mathcal{H}^\mu(x_1) + \sum_{t=1}^{T-1} \gamma^t (r_t + c\mathcal{H}^\mu(x_{t+1})) + \gamma^T Q_{\text{ent}}^\pi(x_T, a_T)] \\ \leq \mathbb{E}_\mu[r_0 + \gamma c\mathcal{H}^\mu(x_1) + \gamma \mathbb{E}_\mu[Q_{\text{ent}}^{\pi_{\text{ent}}^*}(x_1, a_1)]] \\ \leq Q_{\text{ent}}^{\pi_{\text{ent}}^*}(x_0, a_0), \end{aligned}$$

When $\pi = \mu$ we have the special case $\mathbb{E}_\mu[\sum_{t=0}^{\infty} \gamma^t r_t] \leq V^{\pi^*}(x_0)$, the lower bound which motivated the original lower-bound Q-learning based self-imitation learning [8].

B Proof of Theorem 2

For notational simplicity, let $\mathcal{U} := (\mathcal{T}^\mu)^{n-1} \mathcal{T}^\pi$ and let $\tilde{\mathcal{U}}Q(x, a) := Q(x, a) + [UQ(x, a) - Q(x, a)]_+$. As a result, we could write $\mathcal{T}_{\text{n,sil}}^{\alpha, \beta} = (1 - \beta)\mathcal{T}^\pi + (1 - \alpha)\beta\tilde{\mathcal{U}} + \alpha\beta\mathcal{U}$.

First, we prove the contraction properties of $\mathcal{T}_{\beta, \text{n,sil}}^\mu$. Note that by construction $|\tilde{\mathcal{U}}Q_1(x, a) - \tilde{\mathcal{U}}Q_2(x, a)| \leq \max(|Q_1(x, a) - Q_2(x, a)|, |\mathcal{U}Q_1(x, a) - \mathcal{U}Q_2(x, a)|) \leq \|Q_1 - Q_2\|_\infty$. Then through the triangle inequality, $\|\mathcal{T}_{\text{n,sil}}^{\alpha, \beta} Q_1 - \mathcal{T}_{\text{n,sil}}^{\alpha, \beta} Q_2\|_\infty \leq (1 - \beta)\|\mathcal{T}^\pi Q_1 - \mathcal{T}^\pi Q_2\|_\infty + (1 - \alpha)\beta\|\tilde{\mathcal{U}}Q_1 - \tilde{\mathcal{U}}Q_2\|_\infty + \alpha\beta\|\mathcal{U}Q_1 - \mathcal{U}Q_2\|_\infty \leq [(1 - \beta)\gamma + (1 - \alpha)\beta + \alpha\beta\gamma^n]\|Q_1 - Q_2\|_\infty$. This proves the upper bound on the contraction rates of $\mathcal{T}_{\text{n,sil}}^{\alpha, \beta}$. Let $\eta(\alpha, \beta) = (1 - \beta)\gamma + (1 - \alpha)\beta + \alpha\beta\gamma^n$ and set $\eta(\alpha, \beta) < \gamma$, we deduce $\alpha > \frac{1 - \gamma}{1 - \gamma^n}$.

Next, we show properties of the fixed point $\tilde{Q}^{\alpha, \beta}$. This point uniquely exists because $\Gamma(\mathcal{T}_{\text{n,sil}}^{\alpha, \beta}) < 1$ if $(1 - \alpha)\beta < 1$. From $\mathcal{T}_{\text{n,sil}}^{\alpha, \beta} \tilde{Q}^{\alpha, \beta} = \tilde{Q}^{\alpha, \beta}$, we could derive by rearranging terms $(1 - \beta)(\mathcal{T}^\pi \tilde{Q} - \tilde{Q}) + \alpha\beta(\mathcal{U}\tilde{Q} - \tilde{Q}) = -(1 - \alpha)\beta(\tilde{\mathcal{U}}\tilde{Q} - \tilde{Q}) \leq 0$. This further implies that $\mathcal{T}^\pi \tilde{Q} \leq \tilde{Q}$. Now let $\mathcal{T} := \frac{(1 - \beta)}{1 - \beta + \alpha\beta} \mathcal{T}^\pi + \frac{\alpha\beta}{1 - \beta + \alpha\beta} \mathcal{U}$. This simplifies to $\mathcal{T}\tilde{Q} - \tilde{Q} \leq 0$. By the monotonicity of \mathcal{T} , we see $Q^{t\pi + (1-t)\mu^{n-1}\pi} \geq \lim_{k \rightarrow \infty} (\mathcal{T})^k \tilde{Q} = Q^\pi$ where $t = \frac{1 - \beta}{1 - \beta + \alpha\beta}$.

For the another set of inequalities, define $\tilde{H}Q := (1 - \beta)\mathcal{T}^* + (1 - \alpha)\beta\tilde{\mathcal{U}}Q + \alpha\beta(\mathcal{T}^*)^n$, where recall that \mathcal{T}^* is the optimality Bellman operator.

First, note \tilde{H} has Q^{π^*} as its unique fixed point. To see why, let \tilde{Q} be a generic fixed point of \tilde{H} such that $\tilde{H}\tilde{Q} = \tilde{Q}$. By rearranging terms, it follows that $(1 - \beta)(\mathcal{T}^*\tilde{Q} - \tilde{Q}) + \alpha\beta((\mathcal{T}^*)^n\tilde{Q} - \tilde{Q}) = -(1 - \alpha)\beta(\tilde{U}\tilde{Q} - \tilde{Q}) \leq 0$. However, by construction $(\mathcal{T}^*)^i Q \geq Q, \forall i \geq 1, \forall Q$. This implies that $(1 - \beta)(\mathcal{T}^*\tilde{Q} - \tilde{Q}) + \alpha\beta((\mathcal{T}^*)^n\tilde{Q} - \tilde{Q}) \geq 0$. As a result, $(1 - \beta)(\mathcal{T}^*\tilde{Q} - \tilde{Q}) + \alpha\beta((\mathcal{T}^*)^n\tilde{Q} - \tilde{Q}) = 0$ and \tilde{Q} is a fixed point of $t\mathcal{T}^* + (1 - t)(\mathcal{T}^*)^n$. Since $t\mathcal{T}^* + (1 - t)(\mathcal{T}^*)^n$ is strictly contractive as $\Gamma(t\mathcal{T}^* + (1 - t)(\mathcal{T}^*)^n) \leq t\gamma + (1 - t)\gamma^n \leq \gamma < 1$, its fixed point is unique. It is straightforward to deduce that Q^{π^*} is a fixed point of $t\mathcal{T}^* + (1 - t)(\mathcal{T}^*)^n$ and we conclude that the only possible fixed point of \tilde{H} is $\tilde{Q} = Q^{\pi^*}$. Finally, recall that by construction $\tilde{H}Q \geq Q, \forall Q$. By monotonicity, $Q^{\pi^*} = \lim_{k \rightarrow \infty} (\tilde{H})^k \tilde{Q}^{\alpha, \beta} \geq \tilde{Q}^{\alpha, \beta}$. In conclusion, we have shown $Q^{t\pi + (1-t)\mu^{n-1}\mu} \leq \tilde{Q}^{\alpha, \beta} \leq Q^{\pi^*}$.

C Additional theoretical results

Theorem 3. *Let π^* be the optimal policy and V^{π^*} its value function under standard RL formulation. Given a partial trajectory $(x_t, a_t)_{t=0}^n$, the following inequality holds for any n ,*

$$V^{\pi^*}(x_0) \geq \mathbb{E}_\mu \left[\sum_{t=0}^{n-1} \gamma^t r_t + \gamma^n V^{\pi^*}(x_n) \right] \quad (11)$$

Proof. Let π, μ be any policy and π^* the optimal policy. We seek to show $V^{\pi^*}(x_0) \geq \mathbb{E}_\mu[\sum_{t=0}^{T-1} \gamma^t r_t + \gamma^T V^{\pi^*}(x_T)]$ for any $T \geq 1$.

We prove the results using induction. For the base case $T = 1$, $V^{\pi^*}(x_0) = \mathbb{E}_{\pi^*}[Q^{\pi^*}(x_0, a_0)] \geq \mathbb{E}_\mu[Q^{\pi^*}(x_0, a_0)] = \mathbb{E}_\mu[r_0 + \gamma V^{\pi^*}(x_1)] \geq \mathbb{E}_\mu[r_0 + \gamma V^\pi(x_1)]$, where the first inequality comes from the fact that $\pi^*(\cdot | x_0) = \arg \max_a Q^{\pi^*}(x_0, a)$. Now assume that the statement holds for any $T \leq k - 1$, we proceed to the case $T = k$.

$$\begin{aligned} \mathbb{E}_\mu \left[\sum_{t=0}^{k-1} \gamma^t r_t + \gamma^k V^{\pi^*}(x_k) \right] &= \mathbb{E}_\mu \left[r_0 + \gamma \mathbb{E}_\mu \left[\sum_{t=0}^{k-2} \gamma^t r_t + \gamma^{k-1} V^{\pi^*}(x_k) \right] \right] \\ &\leq \mathbb{E}_\mu [r_0 + \gamma V^{\pi^*}(x_1)] \leq V^{\pi^*}(x_0), \end{aligned}$$

where the first inequality comes from the induction hypothesis and the second inequality follows naturally from the base case. This implies that n -step quantities of the form $V^{\pi^*}(x_0) \geq \mathbb{E}_\mu[\sum_{t=0}^{n-1} \gamma^t r_t + \gamma^n V^{\pi^*}(x_n)]$ are lower bounds of the optimal value function $V^{\pi^*}(x_0)$ for any $n \geq 1$. \square

D Experiment details

Implementation details. The algorithmic baselines for deterministic actor-critic (TD3 and DDPG) are based on OpenAI Spinning Up <https://github.com/openai/spinningup> [51]. The baselines for stochastic actor-critic is based on PPO [18] and SIL+PPO [8] are based on the author code base <https://github.com/junhyukoh/self-imitation-learning>. Throughout the experiments, all optimizations are carried out via Adam optimizer [52].

Architecture. Deterministic actor-critic baselines, including TD3 and DDPG share the same network architecture following [51]. The Q-function network $Q_\theta(x, a)$ and policy $\pi_\phi(x)$ are both 2-layer neural network with $h = 300$ hidden units per layer, before the output layer. Hidden layers are interleaved with $\text{relu}(x)$ activation functions. For the policy $\pi_\phi(x)$, the output is stacked with a $\text{tanh}(x)$ function to ensure that the output action is in $[-1, 1]$. All baselines are run with default hyper-parameters from the code base.

Stochastic actor-critic baselines (e.g. PPO) implement value function $V_\theta(x)$ and policy $\pi_\phi(a | x)$ both as 2-layer neural network with $h = 64$ hidden units per layer and tanh activation. The stochastic policy $\pi_\phi(a | x)$ is a Gaussian $a \sim \mathcal{N}(\mu_\phi(x), \sigma^2)$ with state-dependent mean $\mu_\phi(x)$ and a global variance parameter σ^2 . Other missing hyper-parameters take default values from the code base.

Table 1: Summary of the performance of algorithmic variants across benchmark tasks. We use *uncorrected* to denote prioritized sampling without IS corrections. Return-based SIL is represented as SIL with $n = \infty$. For each task, algorithmic variants with top performance are highlighted (two are highlighted if they are not statistically significantly different). Each entry shows mean \pm std performance.

| Tasks | SIL $n = 5$ | SIL $n = 5$ (uncorrected) | SIL $n = 1$ (uncorrected) | 5-step | 1-step | SIL $n = \infty$ |
|----------------|----------------------------------|----------------------------------|-----------------------------------|---------------------------------|-----------------------------------|------------------|
| DMWALKERRUN | 642 \pm 107 | 675 \pm 15 | 500 \pm 138 | 246 \pm 49 | 274 \pm 100 | 320 \pm 111 |
| DMWALKERSTAND | 979 \pm 2 | 947 \pm 18 | 899 \pm 55 | 749 \pm 150 | 487 \pm 177 | 748 \pm 143 |
| DMWALKERWALK | 731 \pm 151 | 622 \pm 197 | 601 \pm 108 | 925 \pm 10 | 793 \pm 121 | 398 \pm 203 |
| DMCHEETAHRUN | 830 \pm 36 | 597 \pm 64 | 702 \pm 72 | 553 \pm 92 | 643 \pm 83 | 655 \pm 59 |
| ANT | 4123 \pm 364 | 3059 \pm 360 | 3166 \pm 390 | 1058 \pm 281 | 3968 \pm 401 | 3787 \pm 411 |
| HALFCHEETAH | 8246 \pm 784 | 9976 \pm 252 | 10417 \pm 364 | 6178 \pm 151 | 10100 \pm 481 | 8389 \pm 386 |
| ANT(B) | 2954 \pm 54 | 1690 \pm 564 | 1851 \pm 416 | 2920 \pm 84 | 1866 \pm 623 | 1884 \pm 631 |
| HALFCHEETAH(B) | 2619 \pm 129 | 2521 \pm 128 | 2420 \pm 109 | 1454 \pm 338 | 2544 \pm 31 | 2014 \pm 378 |

D.1 Further implementation and hyper-parameter details

Generalized SIL for deterministic actor-critic. We adopt TD3 [27] as the baseline for deterministic actor-critic. TD3 maintains a Q-function network $Q_\theta(x, a)$ and a deterministic policy network $\pi_\theta(x)$ with parameter θ . The SIL subroutines adopt a prioritized experience replay buffer: the return-based SIL samples tuples according to the priority $[R^\mu(x, a) - Q_\theta(x, a)]_+$ and minimizes the loss function $[R^\mu(x, a) - Q_\theta(x, a)]_+$; the generalized SIL samples tuples according to the priority $[L^{\pi, \mu, n}(x, a) - Q_\theta(x, a)]_+$ and minimizes the loss function $[L^{\pi, \mu, n}(x, a) - Q_\theta(x, a)]_+$. The experience replay adopts the parameter $\alpha = 0.6, \beta = 0.1$ [53]. Throughout the experiments, TD3-based algorithms all employ $\alpha = 10^{-3}$ for the network updates.

To calculate the update target $L^{\pi, \mu, n}(x_0, a_0) = \sum_{t=0}^{n-1} \gamma^t r_t + Q_{\theta'}(x_n, \pi_{\theta'}(x_n))$ with partial trajectory $(x_t, a_t, r_t)_{t=0}^n$ along with the target value network $Q_{\theta'}(x, a)$ and policy network $\pi_{\theta'}(x)$. The target network is slowly updated as $\theta' = \tau\theta' + (1 - \tau)\theta$ where $\tau = 0.995$ [14].

Generalized SIL for stochastic actor-critic. We adopt PPO [18] as the baseline algorithm and implement modifications on top of the SIL author code base <https://github.com/junhyukoh/self-imitation-learning> as well as the original baseline code <https://github.com/openai/baselines> [54]. All PPO variants use the default learning rate $\alpha = 3 \cdot 10^{-4}$ for both actor $\pi_\theta(a | x)$ and critic $V_\theta(x)$. The SIL subroutines are implemented as a prioritized replay with $\alpha = 0.6, \beta = 0.1$. For other details of SIL in PPO, please refer to the SIL paper [8].

The only difference between generalized SIL and SIL lies in the implementation of the prioritized replay. SIL samples tuples according to the priority $[R^\mu(x, a) - V_\theta(x)]_+$ and minimize the SIL loss function $([R^\mu(x, a) - V_\theta(x)]_+)^2$ for the value function, and $-\log \pi_\theta(a | x)[R^\mu(x, a) - V_\theta(x)]_+$ for the policy. Generalized SIL samples tuples according to the priority $([L^{\pi, \mu, n}(x, a) - V_\theta(x)]_+)^2$, and minimize the loss $([L^{\pi, \mu, n}(x, a) - V_\theta(x)]_+)^2$ and $-\log \pi_\theta(a | x)[L^{\pi, \mu, n}(X, a) - V_\theta(x)]_+$ for the value function/policy respectively.

To calculate the update target $L^{\pi, \mu, n}(x_0, a_0) = \sum_{t=0}^{n-1} \gamma^t r_t + V_{\theta'}(x_n)$ with partial trajectory $(x_t, a_t, r_t)_{t=0}^n$ along with the target value network $V_{\theta'}(x)$. We apply the target network technique to stabilize the update, where θ' is a delayed version of the major network θ and is updated as $\theta' = \tau\theta' + (1 - \tau)\theta$ where $\tau = 0.995$.

D.2 Additional experiment results

Comparison across related baselines. We make clear the comparison between related baselines in Table 1. We present results for n -step TD3 with $n \in \{1, 5\}$; TD3 with generalized SIL with $n = 5$ and its variants with different setups for prioritized sampling; TD3 with return-based SIL ($n = \infty$). We show the results across all 8 tasks - in each entry of Table 1 we show the mean \pm std of performance averaged over 3 seeds. The performance of each algorithmic variant is the average testing performance of the last 10^4 training steps (from a total of 10^6 training steps). The best

Table 2: Comparison between different replay schemes. For each task, algorithmic variants with top performance are highlighted (two are highlighted if they are not statistically significantly different). Each entry shows mean \pm std performance.

| Tasks | SIL $n = 5$ | SIL $n = 5$ (uncorrected) | SIL $n = 5$ (no priority) |
|----------------|----------------------------------|----------------------------------|---------------------------------|
| DMWALKERRUN | 642 \pm 107 | 675 \pm 15 | 424 \pm 127 |
| DMWALKERSTAND | 979 \pm 2 | 947 \pm 18 | 634 \pm 184 |
| DMWALKERWALK | 731 \pm 151 | 622 \pm 197 | 766 \pm 103 |
| DMCHEETAHRUN | 830 \pm 36 | 597 \pm 64 | 505 \pm 182 |
| ANT | 4123 \pm 364 | 3059 \pm 360 | 4358 \pm 496 |
| HALFCHEETAH | 8246 \pm 784 | 9976 \pm 252 | 8927 \pm 596 |
| ANT(B) | 2954 \pm 54 | 1690 \pm 564 | 2910 \pm 88 |
| HALFCHEETAH(B) | 2619 \pm 129 | 2521 \pm 128 | 2284 \pm 85 |

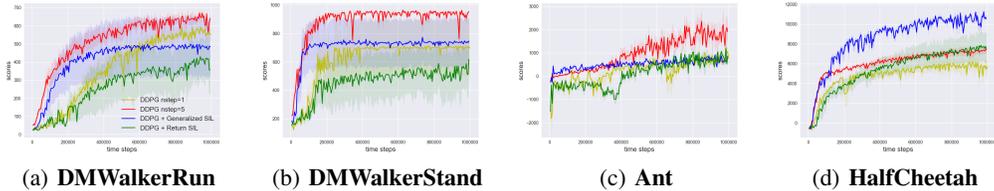


Figure 4: Standard evaluations on 4 simulation tasks for DDPG baselines. Different colors represent different algorithmic variants. Each curve shows the mean \pm 0.5std of evaluation performance during training, averaged across 3 random seeds. The x-axis shows the time steps and the y-axis shows the cumulative returns.

algorithmic variant is highlighted in bold. We see that in general generalized SIL with $n = 5$ performs the best.

Ablation on the prioritized sampling. In prioritized sampling [53], when the tuples $d = (x_i, a_i, r_i)_{i=0}^n \in \mathcal{D}$ are sampled with priorities s_d , it is sampled with probability $p(d) \propto s_d^\alpha$. During updates, the IS correction consists in optimizing the loss $\mathbb{E}_d[w_d l_d]$ where l_d is the loss computed from tuple d and the IS correction weight $w_d = (N \cdot p_d)^{-\beta}$ where N is the number of tuples in the buffer \mathcal{D} .

We compare several prioritized sampling variants of generalized SIL in Table 2. There are three variants: SIL $n = 5$ with both prioritized sampling ($\alpha = 0.6$) and IS correction ($\beta = 0.1$); SIL $n = 5$ with prioritized sampling ($\alpha = 0.6$) only and without IS correction ($\beta = 0.0$); SIL $n = 5$ with no prioritized sampling ($\alpha = \beta = 0.0$). The performance setup in Table 2 is the same as in Table 1. It can be seen from Table 2 that generalized SIL performs the best with full prioritized sampling.

Results on DDPG. DDPG is a baseline actor-critic algorithm with a deterministic actor [15]. Compared to TD3, DDPG does not adopt a double-critic approach [27] and suffers from over-estimation bias of the Q-function [30].

We present the baseline evaluation result of DDPG in Figure 4, where we show the results for a few variants: DDPG with n -step update, $n \in \{1, 5\}$; DDPG with generalized SIL $n = 5$ and DDPG with return-based SIL ($n = \infty$). We see that the performance gains of DDPG with generalized SIL $n = 5$ are not as significant - indeed, overall DDPG with $n = 5$ has the best performance. We speculate that this is partly due to the over-estimation bias of DDPG: the formulation of generalized SIL is motivated by shifting the fixed point Q^π with an positive bias. The baseline algorithm benefits the most from generalized SIL when indeed in practice $Q_\theta \approx Q^\pi$. However, this is not the case for DDPG as the algorithm already has high positive bias in that $Q^\theta > Q^\pi$, which reduces the potential gains that come from generalized SIL.