Nonparametric Policy Improvement in Continuous Action Spaces via Expert Demonstrations

Anonymous authors
Paper under double-blind review

Keywords: Policy Optimization, Policy Improvement, Imitation Learning, Nonparametric methods.

Summary

The policy improvement theorem is a fundamental building block of classical reinforcement learning for discrete action spaces. Unfortunately, the lack of an analogous result for continuous action spaces with function approximation has historically limited the ability of policy optimization algorithms to make large step updates, undermining their convergence speed. Here we introduce a novel nonparametric policy that relies purely on data to take actions and that admits a policy improvement theorem for deterministic Markov Decision Processes (MDPs). By imposing mild regularity assumptions on the optimal policy, we show that, when data come from expert demonstrations, one can construct a nonparametric lower bound on the value of the policy, thus enabling its robust evaluation. The constructed lower bound naturally leads to a simple improvement mechanism, based on adding more demonstrations. We also provide conditions to identify regions of the state space where additional demonstrations are needed to meet specific performance goals. Finally, we propose a policy optimization algorithm that ensures a monotonic improvement of the lower bound and leads to high probability performance guarantees. These contributions provide a foundational step toward establishing a rigorous framework for policy improvement in continuous action spaces.

Contribution(s)

i) We present a novel framework for nonparametric policies on continuous state and action spaces that only requires data coming from expert trajectories.
Context: Modern RL algorithms usually learn a parametrized policy (Schulman et al., 2017), a model of the environment, or both (Hafner et al., 2019; Janner et al., 2019).

ii) Robust policy evaluation: Under mild assumptions on the MDP, we can readily construct a lower bound on the optimal Q-function. Our policy is *greedy* with respect to this bound and surprisingly improves upon it.
Context: The expression for this lower bound ensures that greedy actions can be carried out in closed form, making our policy easy to implement and evaluate. In contrast, standard policy iteration (Sutton & Barto, 2018) relies on computing an (approximate) value function estimate of a policy.

iii) Policy improvement: Our framework leads to a policy improvement mechanism, in which more data yields ever tighter lower bounds. As a result, our policy sequentially improves on the new data.
Context: We provide sufficient conditions for our policy to be *strictly* improving on the new data points. Notably, this method allows for large policy updates, in contrast to policy gradient (Sutton et al., 1999) or trust region methods (Schulman et al., 2015), which take small enough steps to ensure improvement on average.

iv) Policy optimization with guarantees: We present a novel algorithm, inspired by minorization maximization, that monotonically improves our lower value estimate, leading to high probability performance guarantees.
Context: We derive easy-to-check conditions (based on the value function bounds and sampled states) that either guarantee a certain suboptimality or suggest a location where new demonstrations are necessary to meet the performance requirements.

Nonparametric Policy Improvement in Continuous Action Spaces via Expert Demonstrations

Anonymous authors
Paper under double-blind review

Abstract

The policy improvement theorem is a fundamental building block of classical reinforcement learning for discrete action spaces. Unfortunately, the lack of an analogous result for continuous action spaces with function approximation has historically limited the ability of policy optimization algorithms to take large update steps, undermining their convergence speed. Here we introduce a novel nonparametric policy that relies purely on data to take actions and that admits a policy improvement theorem for deterministic Markov Decision Processes (MDPs). By imposing mild regularity assumptions on the optimal policy, we show that, when data come from expert demonstrations, one can construct a nonparametric lower bound on the value of the policy, thus enabling its robust evaluation. The constructed lower bound naturally leads to a simple improvement mechanism, based on adding more demonstrations. We also provide conditions to identify regions of the state space where additional demonstrations are needed to meet specific performance goals. Finally, we propose a policy optimization algorithm that ensures a monotonic improvement of the lower bound and leads to high probability performance guarantees. These contributions provide a foundational step toward establishing a rigorous framework for policy improvement in continuous action spaces.

1 Introduction

The policy improvement theorem is a fundamental result in classical dynamic programming (DP) (Puterman, 1994) and reinforcement learning (RL) (Sutton & Barto, 2018) for discrete action spaces. It guarantees that iterative policy updates lead to performance improvements, underpinning the convergence and optimality of classical algorithms such as policy and value iteration. However, when function approximation is introduced—particularly in continuous action spaces—the intricate relationship between policy parameters and performance outcomes makes it virtually impossible to ensure uniform improvement across all states (Sutton & Barto, 2018).

To address this challenge, research has increasingly focused on policy gradient methods (Williams, 1992), which are particularly well-suited for continuous action spaces (see, e.g., Todorov et al. (2012); Tassa et al. (2018)). Unlike classical approaches that guarantee uniform improvement across all states, policy gradient methods optimize performance *in expectation*. A rich body of work has explored enhancements to these methods, including "natural" policy gradient techniques (Peters & Schaal, 2008), methods that aim for monotonic improvement (in expectation) through constrained approximate policy iteration (Schulman et al., 2015), and approaches that take multiple small steps per data batch toward better performance (Schulman et al., 2017). Despite their advantages, these approaches often suffer from slow convergence, sensitivity to hyperparameter tuning, and instability.

This paper presents a novel nonparametric policy improvement mechanism as a viable alternative to policy optimization in problems with continuous state and action spaces. Establishing a policy improvement theorem in this setting would enable large policy updates while maintaining a guarantee of strict improvement. Naturally, achieving such a result requires overcoming the challenges

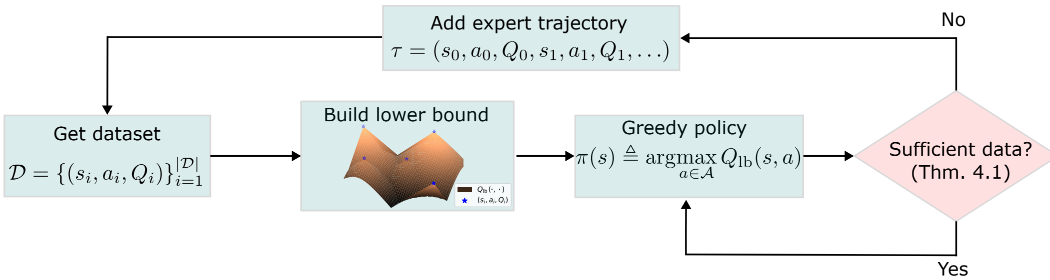

Figure 1: Overview of the proposed method. From left to right: *i)* a dataset containing expert triplets (s_i, a_i, Q_i) is used to *ii)* build a lower bound on the optimal value function; *iii)* acting greedily with respect to it gives our policy; *iv)* if high-probability suboptimality conditions are not met, we collect more expert trajectories and repeat the process.

posed by the intricate dependence between policy parameters and MDP performance. We address this by carefully designing the policy representation and leveraging a minorization-maximization (MM) approach, similar to MM algorithms (Ortega & Rheinboldt, 2000; Sun et al., 2016), to ensure strict improvement over a lower bound of the policy value.

Contributions: The contributions of this work are listed next. For a more detailed discussion on the placement of our work in the literature, we refer the reader to Section 6.

- *Nonparametric Policy Evaluation:* We introduce a novel policy representation for continuous state-action spaces that relies purely on data, i.e., it is nonparametric. We show that under minor regularity assumptions on the optimal policy $\pi^\star$, the proposed policy π admits nonparametric lower estimates $V_{\text{lb}}(s)$ and $Q_{\text{lb}}(s, a)$ of the policy value $V^\pi(s)$ and action value $Q^\pi(s, a)$.
- *Policy Improvement Theorem:* Combining the proposed policy representation and lower bound estimation naturally leads to a policy improvement mechanism that requires only a properly chosen expert trajectory. We provide further conditions on the dataset and the new trajectory that guarantee strict improvement over a region of the state space.
- *Suboptimality Gap and Active Sampling:* While in principle, any expert demonstration would lead to better performance, our analysis derives suboptimality conditions (based on the initial states and bounds on the optimal value function) that either guarantee a certain level of performance or suggest a new location where new expert trajectories are necessary to meet the performance requirements.
- *Nonparametric Policy Optimization:* The aforementioned results lead to a novel algorithm, inspired by minorization-maximization, that monotonically improves our performance lower estimate $V_{\text{lb}}(s)$, leading to high probability performance guarantees, while limiting the amount of data that needs to be stored.

2 Problem setup

We consider a Markov Decision Process $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, T, \rho, \gamma \rangle$ with state space $\mathcal{S}$, action space $\mathcal{A}$, reward set $\mathcal{R}$, initial state distribution ρ, discount factor $\gamma \in (0, 1)$ and transition density $T(s, a, s')$ (Van Hasselt & Wiering, 2007). As usual, policies $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ map states to probability distributions over the action space.[1] Given a policy π, its *value function* and *action-value function* can be defined at any state as:

$$V^\pi(s) \triangleq \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s \right]$$

[1]For deterministic policies, we abuse notation and let $\pi : \mathcal{S} \to \mathcal{A}$, that is to say: $\pi(s_t) = a_t$.

$$Q^\pi(s,a) \triangleq \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t) \mid s_0 = s, a_0 = a\right],$$

where $r(s,a) = \mathbb{E}\left[r_{t+1} \mid s_t = s, a_t = a\right]$ and $\mathbb{E}_\pi[\,\cdot\,]$ denotes expectation with respect to trajectories induced by the MDP and policy π (Sutton & Barto, 2018). The optimal value- and action-value functions are defined for all $s \in \mathcal{S}$, $a \in \mathcal{A}$:

$$V^\star(s) = \max_\pi V^\pi(s); \qquad\qquad Q^\star(s,a) = \max_\pi Q^\pi(s,a).$$

We let $\pi^\star$ stand for the optimal policy, i.e., the maximizer of the two expressions above. A usual goal in RL is to find said policy. A related but usually simpler one is to find:

$$\max_\pi \mathbb{E}_{s\sim\rho}\left[V^\pi(s)\right],$$

that is to say, a policy that is optimal with respect to the initial state distribution ρ. For further discussions on optimality with respect to an initial state distribution, see, e.g., Puterman (1994).

Additional assumptions We make the following assumptions on the MDP and the optimal value function.

Assumption 2.1 (Deterministic MDP)**.** The transition map is deterministic: i.e. there exists $f : \mathcal{S}\times\mathcal{A}\to\mathcal{S}$ such that $s_{t+1} = f(s_t,a_t)$.

Assumption 2.2 ($Q^\star$ is Lipschitz)**.** The optimal action-value function $Q^\star$ is L-Lipschitz, that is:

$$|Q^\star(s,a) - Q^\star(s',a')| \le L\left(\|s-s'\| + \|a-a'\|\right)$$

$\forall s, s' \in \mathcal{S}$ and $\forall a, a' \in \mathcal{A}$.

As we will see shortly, having a Lipschitz optimal value function will allow us to readily compute lower bounds (provided L is known). As it turns out, if $Q^\star$ is Lipschitz so is $V^\star$.

Proposition 2.3. *If $Q^\star$ is L-Lipschitz then $V^\star$ is L-Lipschitz:*

$$|V^\star(s) - V^\star(s')| \le L\|s-s'\| \quad \forall s, s' \in \mathcal{S}.$$

Proof. The proof is in Supplementary Material A.1. □

Assumption 2.2 is not overly restrictive and has been made before (Buşoniu et al., 2018; Shen & Yang, 2021). We present conditions on the MDP that are sufficient to guarantee it.

Proposition 2.4 (Sufficient conditions for Lipschitz value functions (Buşoniu et al., 2018))**.** *If the transition map f and rewards r are Lipschitz, i.e.:*

$$\|f(s,a) - f(s',a')\| \le L_f\left(\|s-s'\| + \|a-a'\|\right)$$
$$|r(s,a) - r(s',a')| \le L_r\left(\|s-s'\| + \|a-a'\|\right)$$

for positive scalars L_f, L_r, and the discount factor satisfies $\gamma L_f < 1$, then $Q^\star$ and $V^\star$ are L-lipschitz with $L \le \frac{L_r}{1-\gamma L_f}$.

Proof. The proof is presented in Supplementary Material A.2 for completeness. □

Our last assumption is related to the data available to the agent, which must come from *expert demonstrations*.

Assumption 2.5 (Expert data)**.** Our agent has access to a collection of triplets[2] $\mathcal{D} = \{(s_i,a_i,Q_i)\}_{i=1}^{|\mathcal{D}|}$ where the state-action pairs are induced by $\pi^\star$ and $Q_i \equiv Q^\star(s_i,a_i)$.

This last assumption on expert data will allow us to state suboptimality results with respect to the optimal policy. It, however, can be relaxed to data collected by any other policy, as long as its value function is Lipschitz. We postpone further comments on this relaxation until the end of Section 3.

[2]We use $\mathcal{D}$ to denote *dataset*: this will be the data that our policy leverages.

Bounds on the optimal value functions We use the fact that $Q^\star$ is Lipschitz (Assumption 2.2) to construct lower bounds on both $V^\star$ and $Q^\star$. These bounds are defined with respect to the information provided in the dataset $\mathcal{D}$.

$$V_{\mathrm{lb}}(s) \triangleq \max_{1\leq i\leq|\mathcal{D}|} \{Q_i - L\|s - s_i\|\}, \tag{1}$$

$$Q_{\mathrm{lb}}(s,a) \triangleq \max_{1\leq i\leq|\mathcal{D}|} \big\{Q_i - L\big(\|s - s_i\| + \|a - a_i\|\big)\big\}. \tag{2}$$

We can, in a similar way, define upper bounds:

$$V_{\mathrm{ub}}(s) \triangleq \min_{1\leq j\leq|\mathcal{D}|} \{Q_j + L\|s - s_j\|\}, \tag{3}$$

$$Q_{\mathrm{ub}}(s,a) \triangleq \min_{1\leq j\leq|\mathcal{D}|} \big\{Q_j + L\big(\|s - s_j\| + \|a - a_j\|\big)\big\}. \tag{4}$$

We omit the dependence of these bounds on $\mathcal{D}$ to avoid clutter. Since both value functions are Lipschitz, the quantities defined above indeed serve as lower and upper bounds (hence the subscripts lb and ub) to the optimal state- and action-value function, respectively:

$$V_{\mathrm{lb}}(s) \leq V^\star(s) \leq V_{\mathrm{ub}}(s) \qquad\qquad Q_{\mathrm{lb}}(s,a) \leq Q^\star(s,a) \leq Q_{\mathrm{ub}}(s,a).$$

Combining upper and lower bounds (in particular for $V^\star$) will come in handy to derive suboptimality guarantees of our policy. We pay special attention to the lower bounds, which will be used to define our nonparametric policy and which we adress in the following section.

3 Nonparametric policies

In this section, we build on the lower bounds introduced in the prequel and propose our nonparametric policy. There are three main ingredients to this construction (highlighted in Figure 1). First, given a dataset $\mathcal{D}$, we construct the lower bounds (1) and (2). We then define a policy that acts greedily with respect to this lower bound. Remarkably, we show that *the value function of this policy improves upon the lower bound*. Let us first start by defining the policy.

Definition 3.1 (Nonparametric policy)**.** For every state $s \in \mathcal{S}$ we define:

$$\pi(s) \triangleq \arg\max_{a\in\mathcal{A}} Q_{\mathrm{lb}}(s,a)$$

As we highlighted before, π acts *greedily* with respect to the lower bound. Notably, this maximization is simple to carry out and always gives actions in the dataset.

Remark 3.2. $\pi(\cdot)$ always chooses an action from the dataset, ie:

$$\forall s \in \mathcal{S}: \; \pi(s) = a_i \text{ for some } i \in \{1, \ldots, |\mathcal{D}|\}.$$

If multiple maximizers exist for a given s, we choose the a_i with the smallest index i, rendering our policy deterministic. If we let $i^\star$ be the maximizer for a given (s,a) pair, notice that we have:

$$Q_{\mathrm{lb}}(s,a) \leq Q_{\mathrm{lb}}(s,a_{i^\star}) = V_{\mathrm{lb}}(s) = Q_{i^\star} - L\|s - s_{i^\star}\|.$$

Policy interpretation Our policy acts in two steps. First, it selects the index $i^\star$ that maximizes $V_{\mathrm{lb}}(s)$ in (1), which amounts to performing a biased projection onto states in the dataset, with bias terms given by Q_i/L. Then, it selects the action $a_{i^\star}$, corresponding to the projected state. Because of the first step, our method bears resemblance to nearest neighbor approaches in RL (Santamaria et al., 1997; Shah & Xie, 2018).

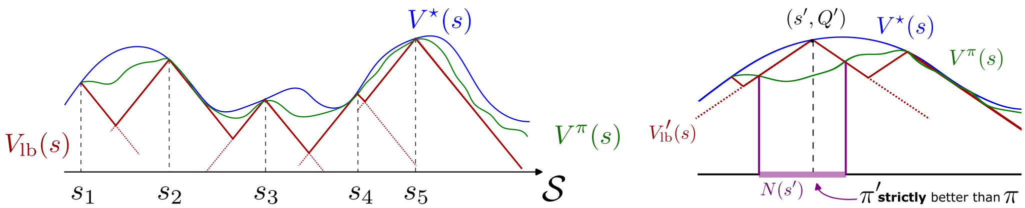

Figure 2: Illustrations of Theorems 3.4 and 3.5. *Left*: Robust policy evaluation. V^π lies between V_{lb} and $V^\star$; all three functions interpolate the data points (s_i, Q_i). *Right*: Policy improvement. adding the transition (s', a', Q') yields a better lower bound $V_{\text{lb}} \leq V'_{\text{lb}}$. Furthermore, *strict* policy improvement holds in the neighborhood $N(s')$.

Greedy policies are ubiquitous in the RL literature (Sutton & Barto, 2018; Williams & Baird, 1993). Since they enable policy improvement, they serve as one of the fundamental building blocks for policy iteration methods (Sutton & Barto, 2018; Pirotta et al., 2013). We will soon show that our policy satisfies a policy evaluation inequality, and that—sequentially—adding more data to the dataset $\mathcal{D}$ yields a form of policy improvement.

Our result will hinge on the fact that the expert data comes from *trajectories*. To that end, we make the last definition before our main results.

Definition 3.3 (Consistent dataset)**.** $\mathcal{D}$ is a consistent dataset if for all $(s_i, a_i, Q_i) \in \mathcal{D}$ the following two conditions hold:

i) $a_i = \pi^\star(s_i)$; $Q_i = V^\star(s_i)$.

ii) $\exists (s_j, a_j, Q_j) \in \mathcal{D}$ such that $s_j = f(s_i, a_i)$.

A dataset made up of expert trajectories[3] of the form $\tau^k = (s_0^k, a_0^k, Q_0^k, s_1^k, a_1^k, Q_1^k, \dots)$ satisfies the consistency definition above.

Policy evaluation and improvement One of our key finding is that the greedy policy defined above has a value function that improves upon the lower bound of the optimal one. We state this result next.

Theorem 3.4 (Policy evaluation)**.** *Let $\mathcal{D}$ be a consistent dataset (Definition 3.3) and π as in Definition 3.1. Then, for all $s \in \mathcal{S}$ the following two inequalities hold:*

$$V_{\text{lb}}(s) \leq r(s, \pi(s)) + \gamma V_{\text{lb}}(f(s, \pi(s))$$
$$V_{\text{lb}}(s) \leq V^\pi(s) \leq V^\star(s) \ .$$

Proof. The proof is in Supplementary Material A.3. □

We want to stress the relevance of the second inequality above, which is depicted in Figure 2 (to the left). In standard policy iteration algorithms (Sutton & Barto, 2018), one first *evaluates* a given policy, resulting in a value function, and then acts greedily upon it. Notably, we act greedily with respect to Q_{lb}, which *may not correspond to the value of any policy*, and still improve upon it. Next, if our greedy policy surpasses this lower bound, the natural thing to do is to increase the size of $\mathcal{D}$ to get a better lower bound. This leads to the policy improvement mechanism highlighted next.

Theorem 3.5 (Policy improvement)**.** *Let $\mathcal{D}$, $\mathcal{D}'$ be consistent datasets with $\mathcal{D} \subset \mathcal{D}'$. Let V_{lb} and V'_{lb} be the lower bounds constructed with $\mathcal{D}$ and $\mathcal{D}'$ respectively. Then the following* ***non-deterioration*** *conditions hold:*

$$V_{\text{lb}}(s) \leq V'_{\text{lb}}(s) \quad \forall s \in \mathcal{S}, \ \textit{and}$$

[3] Although the RL objective pertains infinite-length trajectories, in practice we will truncate them after a horizon $H \geq (1-\gamma)^{-1}$.

$$V^{\pi}(s) \leq V^{\pi'}(s) \quad \forall s \in \Pi_{\mathcal{S}}[\mathcal{D}' \backslash \mathcal{D}],$$

where $\Pi_{\mathcal{S}}[\mathcal{D}] \triangleq \{s_i : \exists a_i, Q_i \text{ such that } (s_i, a_i, Q_i) \in \mathcal{D}\}$ *and "*$\backslash$*" denotes set difference. Furthermore, if there exists* $s' \in \Pi_{\mathcal{S}}[\mathcal{D}' \backslash \mathcal{D}]$ *and an open ball* $\mathcal{B}(s')$ *such that* $\sup_{s \in \mathcal{B}(s')} V^{\pi}(s) < V^{\star}(s')$*, then* ***strict improvement*** *exists in a subset* $N(s') \subset \mathcal{B}(s')$*:*

$$\begin{aligned} V_{\text{lb}}(s) &< V'_{\text{lb}}(s) \quad \forall s \in N(s'), \text{ and} \\ V^{\pi}(s) &< V^{\pi'}(s) \quad \forall s \in N(s') \end{aligned}$$

Proof. The proof is in Supplementary Material A.4 □

By refining the lower bounds on $V^{\star}$, we can improve the value of our policy, specifically on new transitions. However, in general we cannot claim (like in classical policy iteration) $V^{\pi}(s) \leq V^{\pi'}(s)$ uniformly over $s \in \mathcal{S}$, nor even uniformly over the initial state distribution, that is to say: $\mathbb{E}_{s\sim\rho}[V^{\pi}(s)] \leq \mathbb{E}_{s\sim\rho}[V^{\pi'}(s)]$. This is typical of majorization-minimization methods (like ours) that perform sequential optimization with respect to an improved lower bound (Ortega & Rheinboldt, 2000; Sun et al., 2016).

Notwithstanding, the hope is that refinements of the lower bounds—attained by adding new trajectories to the dataset $\mathcal{D}$—will improve the performance of our resulting policy π. Notably, we derive easy-to-check, sufficient conditions to achieve an ε-suboptimality that we address next. After defining these suboptimality notions and the conditions that will attain them, we will be ready to present our algorithm.

On suboptimality and guarantees We measure the suboptimality of our policy with the gap between V^{π} and $V^{\star}$.

Definition 3.6 (Suboptimality)**.** Let $\varepsilon \geq 0$. We say π is ε-suboptimal whenever, for all $s \in \mathcal{S}$:

$$V^{\star}(s) - V^{\pi}(s) \leq \varepsilon.$$

If the dataset *covers* the state space $\mathcal{S}$ in a sense to be made explicit, then our resulting policy will have the desired suboptimality.

Proposition 3.7 (Suboptimality guarantee)**.** *Let* $V_{\text{lb}}(s)$ *and* $V_{\text{ub}}(s)$ *be the* $\mathcal{D}$*-dependent lower and upper bounds of* $V^{\star}$ *defined in* (1)*,* (3)*. If for every* $s \in \mathcal{S}$ *we have:*

$$\text{SurrogateGap}(s) \triangleq V_{\text{ub}}(s) - V_{\text{lb}}(s) \leq \varepsilon, \tag{5}$$

Then π *is* ε*-suboptimal.*

Proof. The proof is in Supplementary Material A.5 □

Notice that computing (5) for a fixed s is simple, since both the upper and lower bounds can be computed by maximizing over states in $\mathcal{D}$. This gap—which overestimates the gap of the policy—decreases as more data is added to $\mathcal{D}$. In the next section, we present an algorithm that checks this condition at the start of each episode. This will inform our agent when it needs to collect more expert data from the environment. Since it is infeasible to check the condition of Theorem 3.7 for the whole state space, we will come up with high probability guarantees (with respect to the initial state distribution ρ) to achieve a desired threshold.

Algorithm 1: NPP: NonParametric Policy

Input: $L > 0.$; `/* Lipschitz constant */`
1 $\varepsilon > 0$; `/* Suboptimality gap */`
2 Function `TrajectoryOptimizer(·)` ; `/* Call to gather expert data */`
3 Suboptimality condition: *i)* or *ii)* in Theorem 4.1.
Output: A policy π satisfying Thm 4.1 .
4 **Initialize:** $\mathcal{D} = \emptyset$
5 **for** *each episode* e*=1,...* **do**
6     $s \sim \rho$; `// Reset environment`
7     $\Delta_e = \text{SurrogateGap}(s)$; `// Over-estimator of gap (5)`
8   **if** $\Delta_e \leq \varepsilon$ **then**
        `// Policy is good enough`
9   **continue**
10   **else**
        `// Need more data`
11         $\tau = (s_0, a_0, V_0^\star, \ldots, S_{H-1}) =$ `TrajectoryOptimizer`(s)
12   **for** $i = 0, \ldots, H-1$ **do**
13             $\mathcal{D}$.append$\big((s_i, a_i, Q_i)\big)$; `// Add transitions to dataset`
14   **end**
15   **if** *Condition in Thm. 4.1 holds for* $[\Delta_e,\ \Delta_{e-1}, \ldots \Delta_{e-n+1}]$ **then**
        `// Policy is approximately optimal w.h.p.`
16   **break**
17 **end**

What if the data is suboptimal? Our main results in the preceeding sections relied on data coming from an expert or optimal policy. In practical applications of behavioral cloning (Torabi et al., 2018; Florence et al., 2022) or imitation learning (Hussein et al., 2017a; Osa et al., 2018b) this is seldom the case. We can relax this assumption. As long as the data comes from a policy with a Lipschitz value function, we can construct the lower bounds and still improve upon them. Further discussion on these evaluation/improvement results are in Supplementary Material B, along with experiments to support it.

4 Algorithm

Theorems 3.4 and 3.5 presented in Section 3 pave the way to Algorithm 1. Given a dataset $\mathcal{D}$, our policy constructs V_{lb} and then acts greedily with respect to that lower bound. If more data is required, we call a `TrajectoryOptimizer`, generate a new trajectory and use it to build a new dataset $\mathcal{D}' \supset \mathcal{D}$. The algorithm terminates whenever it can guarantee (with high probability) that a suboptimality condition is met. In Theorem 3.7 we state sufficient conditions—required on the whole state space—to achieve said suboptimality. We now present a finite-sample analysis based on episodic metrics that will enable us to state that the suboptimality has been achieved with high-probability.

Guarantees We want to analyze the performance of policy π coming out of Algorithm 1 after running it for E rounds. Since it is infeasible to check the condition of Theorem 3.7 on the whole initial state distributions, we will derive sample complexity bounds that guarantee either $\mathbb{E}_{s\sim\rho}\left[V^\star(s) - V^\pi(s)\right] \leq \varepsilon$ or $\mathbb{P}_{s\sim\rho}\left[V^\star(s) - V^\pi(s) \leq \varepsilon\right]$ with high probability.

Theorem 4.1 (Probabilistic Guarantees)**.** *Assume Algorithm 1 ran for E episodes; let Δ_e be defined as in line 7 of the algorithm. Let $\mathcal{S}_0$ denote the support of the initial state distribution ρ.*

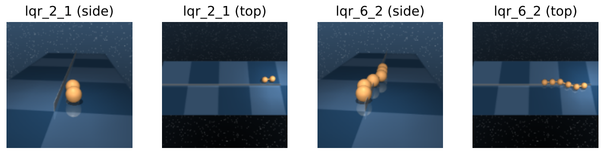

Figure 3: The `lqr` environments from DeepMind Control suite.

i) If for the last n episodes no new data has been collected, then with probability at least $1-\delta$, we have:

$$\mathbb{P}_{s\sim\rho}\left[V^\star(s) - V^\pi(s) \leq \varepsilon\right] \geq p,$$

provided:

$$n \geq \frac{1}{1-p} \log \frac{1}{\delta}\,.$$

ii) Let $\bar{\Delta}_n \triangleq \frac{1}{n}\sum_{i=0}^{n-1} \Delta_{E-i}$. Then with probability at least $1-\delta$ we have:

$$\mathbb{E}_{s\sim\rho}\left[V^\star(s) - V^\pi(s)\right] \leq \varepsilon,$$

provided:

$$\bar{\Delta}_n < \varepsilon \quad \textit{and} \quad n \geq \frac{2L^2\,\mathrm{diam}^2(\mathcal{S}_0)}{(\varepsilon - \bar{\Delta}_n)^2} \log \frac{1}{\delta}\,.$$

Proof. The proof is in Supplementary Material A.6 □

The algorithm takes as input one of these suboptimality notions—either having low probability of exceeding the gap, or satisfying the gap in expected value—and terminates whenever the conditions of the preceeding theorem are satisfied.

5 Experiments

In this section, we show the performance of Algorithm 1 on two LQR environments. In these settings, the optimal policy and the optimal value function exist in closed form, yielding a convenient way of computing expert trajectories.

Environments We test our algorithm on environments from the `DeepMind Control suite` (Tassa et al., 2018; Tunyasuvunakool et al., 2020), which are based on the `MuJoCo` engine (Todorov et al., 2012). The `lqr_n_m` environments are shown in Figure 3. They constitute a well-studied problem in control theory with a closed form solution for the optimal policy and value function (Bertsekas, 2012). This available optimal policy serves as the trajectory optimizer of Algorithm 1.

The environments are made up of a body of n balls in series attached by strings, the first m of which are actuated, i.e. $\dim(\mathcal{A}) = m$. The balls move along one axis, positions and velocities yield a state vector of $\dim(\mathcal{S}) = 2n$. The goal in `lqr` is to bring the system close to the origin, with stage reward $r(s,a) = 1 - 0.5(\|s\|^2 + 0.1\cdot\|a\|^2)$. Originally, an episode terminates whenever $\|s\| \leq 10^{-6}$. Initial states have zero velocity and the n positions are sampled uniformly from a sphere of radius $\sqrt{2}$.

We perform systematic evaluation of these two environments under the optimal policy to come up with upper bounds on the Lipschitz constant of the optimal value functions, and to fix the horizon for each environment. We ended using $L = 50$ for `lqr_2_1` and $L = 200$ for `lqr_6_2`. The horizon for both environments is set to $H = 1000$. See Figures 8 and 9 in Supplementary Material C for further details. If the Lipschitz constant is not known beforehand, it can be estimated based on the dataset, either globally (using the whole data) or locally, by using k-nearest neighbors of a query point s.

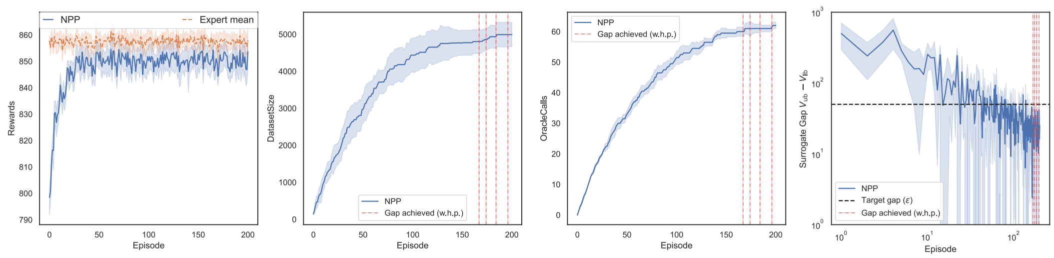

Figure 4: Training curves for `lqr_2_1` with target suboptimality $\varepsilon = 50$, with results averaged over 4 seeds. *Left*: Episodic return of policy π (in blue) and expert (in orange) at different stages of training. $N = 100$ rollouts are performed at each point; solid line corresponds to the median and shaded area to a 95% confidence interval. *Middle-left*: size of the dataset. *Middle-right*: calls to the `TrajectoryOptimizer` oracle (notice calls are made on approximately 30% of the episodes). *Right*: surrogate gap $V_{\text{ub}} - V_{\text{lb}}$ for the initial states. Purple dashed lines correspond to the hitting times (one per seed) for reaching the target suboptimality gap.

Results on `lqr_2_1` We set a target a suboptimality gap $\varepsilon = 50$ (which corresponds to being 6% away from the optimal policy) and choose the probabilistic guarantee given by condition *(i)* of Theorem 4.1 with $p = 0.9, \delta = 0.1$. Training curves are in Figure 4, showing the (evaluation) return of our policy against the optimal one, the size of the dataset, the number of calls to the oracle (the trajectory optimizer) and the $\mathrm{SurrogateGap}$ defined in (5). As seen in the rightmost plot, prior to the 200$^{\text{th}}$ episode, the surrogate gap is below ε for consecutive episodes so as to satisfy condition *(i)* in Thm. 4.1. Judging by the leftmost plot, this suboptimality is reached much sooner. These results are supported by Figure 5, where we show (minus) the empirical suboptimality distribution $V^{\pi} - V^{\star}$ from random initial states $s_0 \sim \rho$, with $N = 100$ rollouts per episode. One can see that, when the algorithm terminates, the gap is bounded by ε as desired. The densities were constructed using KDE (Rosenblatt, 1956; Chen, 2017). Additional plots showing the distribution of states in the dataset, see Supplementary Material D.

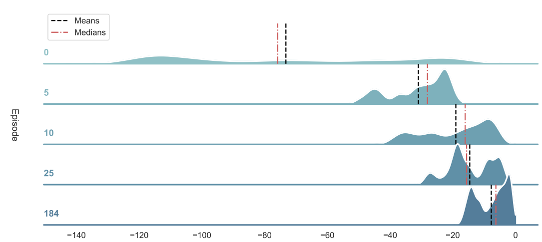

Figure 5: Distribution of the suboptimality gap $V^{\pi}(s_0) - V^{\star}(s_0), s_0 \sim \rho$ at different stages of training on `lqr_2_1`. See the caption in Figure 4 for experiment details. The gap shrinks as training progresses, and at the last episode our algorithm certifies with high probability the desired gap of $\varepsilon = 50$.

Results on `lqr_6_2` For this environment we set a gap of $\varepsilon = 300$, and, like before, the probabilistic requirements of condition *(i)* in Thm. 4.1, with $\delta = 0.1$ and $p = 0.9$. Training curves and suboptimality distribution estimates are in Figure 6. As can be seen in the leftmost and rightmost plot, that show the "true" gap in value, we reach the desired suboptimality in less than 1000 steps. The certification based on the surrogate gap (third plot) was not achieved at this mark, but the trend suggests it would happen with longer runtime. Our guarantees are conservative, but the algorithm reaches the desired results suprisingly much faster.

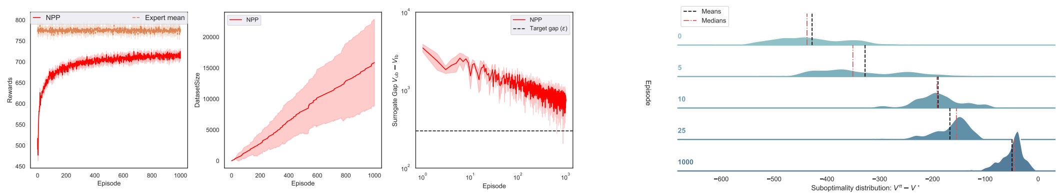

Figure 6: Training curves and suboptimality distribution for `lqr_6_2`. See the caption in Figure 4 for details on how to read the plots.

6 Related Work

Policy Improvement and Related Classical Algorithms. The Policy Improvement Theorem is attributed to Richard Bellman in the 1950s and first appeared in Bellman (1957). Policy Iteration (PI), which leverages the Policy Improvement Theorem to iteratively obtain uniformly better policies, is due to Howard (1960). PI requires that at each iteration, a policy is (approximately) evaluated, which sometimes is construed as computationally costly, even in discrete spaces. Value Iteration (VI) was introduced by MacQueen (1966) and later extended by Van Nunen (1976) as an alternative method that does not require policy evaluation. Notably, the majority of these algorithms have classical extensions for function approximation; see, e.g., Bertsekas (1996) for a thorough discussion of all these methods. However, such methods are either limited to discrete action spaces or lack convergence guarantees and often fail to converge (Bertsekas, 2011). Our framework, which is naturally applicable to settings with continuous action spaces, shares commonalities with both VI and PI. As in the case of VI, Algorithm 1, VI iteration constructs a sequence of monotonically increasing functions (V_{lb}) that lead to increasingly better lower estimates for $V^\star$. However, our algorithm also guarantees that V_{lb} is a lower bound for V^π (Theorem 3.4). Similarly, akin to PI, our results provide guarantees for non-deterioration (of the lower bound V_{lb}) and strict improvement of V^π on some region of the state space.

Nonparametric Methods in Reinforcement Learning. Nonparametric methods have been extensively studied in reinforcement learning (RL), with applications ranging from value function approximation to policy optimization. Traditional approaches often rely on nearest neighbor regression (Santamaria et al., 1997; Shah & Xie, 2018; McCallum, 1994), and kernel-based techniques (Ormoneit & Sen, 2002; Domingues et al., 2021), for *nonparametric policy evaluation*, where function approximation is used to estimate value some policy. These methods typically fit a value function (Q or V) and derive a policy through greedy optimization over the estimated function. However, a key limitation of these approaches is their reliance on value function estimation, which can be sensitive to approximation errors and data sparsity. In contrast, our method does not attempt to estimate the value function but instead constructs a global lower bound on the policy value. *Nonparametric policies*, akin to the ones proposed in this paper, has been proposed in the past. In particular, several works have consider the use of nearest neighbor policies, (Mansimov & Cho, 2018; Alton & van de Panne, 2005; Sharon & van de Panne, 2005). However, such methods do not consider a lower estimate on the value of the function when selecting the action. As a result, such methods lack theoretical guarantees on the achievable performance, a key feature of the proposed work. A recent work by Shen & Yang (2021) is most similar to ours, although authors here use nearest neighbors to construct an optimistic overapproximation of the Q function. Their method, in contrast, does not have an easy closed form solution for greedy actions with respect to that bound, instead they use this approximation in an actor-critic framework.

Imitation Learning. Our method is related to, but distinct from, existing approaches to imitation learning (IL; (Argall et al., 2009; Hussein et al., 2017b; Osa et al., 2018a)). IL, or learning from demonstration, seeks to mimic the behavior of an expert in a sequential decision-making problem. Early neural-network-based approaches (Pomerleau, 1988; Schaal, 1996; Atkeson & Schaal, 1997) focused on behavioral cloning for robotics. To address distribution shift between training and deployment, methods were introduced (Ross et al., 2011; Ross & Bagnell, 2014) that query the expert on states encountered by the agent throughout training. Adversarial frameworks (Ho & Ermon, 2016) were found to improve policy robustness in some circumstances. Recently more expressive policy classes, including diffusion models (Chi et al., 2024), have been applied to capture multimodal decision-making in the data. Like these methods, our approach seeks to replicate the performance of an expert given a static dataset. However, it differs fundamentally from these works in being nonparametric.

7 Conclusion

In this work we introduced foundations for policy improvement in continuous action spaces via a nonparametric policy representation that admits a policy improvement theorem. By leveraging expert demonstrations, we provided a principled approach to evaluating and improving policies through a lower-bound estimation of their value. Our results highlight conditions under which additional demonstrations are necessary to ensure performance guarantees, leading to a novel policy optimization algorithm with monotonic improvement properties.

Future work includes extending our theoretical framework to stochastic MDPs, exploring practical implementations in high-dimensional control tasks, and investigating sample efficiency trade-offs in real-world applications. Additionally, refining the proposed algorithm in a setting where the Lipschitz constant is unknown could further enhance its applicability in various domains.

References

Ken Alton and Michiel van de Panne. Learning to steer on winding tracks using semi-parametric control policies. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 4588–4593. IEEE, 2005.

Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2008.10.024.

Christopher G. Atkeson and Stefan Schaal. Robot learning from demonstration. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pp. 12–20, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc. ISBN 1558604863.

Richard Ernest Bellman. *Dynamic programming.* Princeton University Press, Princeton, 1957.

Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*, volume 4. Athena scientific, 2012.

Dimitri Bertsekas. *Reinforcement learning and optimal control*, volume 1. Athena Scientific, 2019.

Dimitri P Bertsekas. Approximate policy iteration: A survey and some new methods. *Journal of Control Theory and Applications*, 9(3):310–335, 2011.

DP Bertsekas. Neuro-dynamic programming. *Athena Scientific*, 1996.

Lucian Buşoniu, Elődi Páll, and Rémi Munos. Continuous-action planning for discounted infinite-horizon nonlinear optimal control with lipschitz values. *Automatica*, 92:100–108, 2018.

Yen-Chi Chen. A tutorial on kernel density estimation and recent advances. *Biostatistics & Epidemiology*, 1(1):161–187, 2017.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion, 2024. URL `https://arxiv.org/abs/2303.04137`.

Omar Darwiche Domingues, Pierre Ménard, Matteo Pirotta, Emilie Kaufmann, and Michal Valko. Kernel-based reinforcement learning: A finite-time analysis. In *International Conference on Machine Learning*, pp. 2783–2792. PMLR, 2021.

Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on Robot Learning*, pp. 158–168. PMLR, 2022.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 4572–4580, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Ronald A Howard. Dynamic programming and markov processes. *MIT Press*, 2:39–47, 1960.

Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017a.

Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: a survey of learning methods. *ACM computing surveys*, 50, 2017b. ISSN 0360-0300. doi: 10.1145/3054912. URL http://hdl.handle.net/10059/2298. COMPLETED – Issue in progress, but article details complete; not marked as pending following upload 10.05.2017 GB – Now on ACM website, issue still in progress 9/5/2017 LM – Not on journal website 24/2/2017 LM – Info from contact 10/2/2017 LM ADDITIONAL INFORMATION: Elyan, Eyad – Panel B.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32, 2019.

Michael J Kearns and Umesh Vazirani. *An introduction to computational learning theory*. MIT press, 1994.

J MacQueen. A modified dynamic programming method for markovian decision problems. *Journal of Mathematical Analysis and Applications*, 14(1):38–43, 1966.

Elman Mansimov and Kyunghyun Cho. Simple nearest neighbor policy method for continuous control tasks. *Open Review*, 2018.

R Andrew McCallum. Instance-based state identification for reinforcement learning. *Advances in Neural Information Processing Systems*, 7, 1994.

Dirk Ormoneit and Śaunak Sen. Kernel-based reinforcement learning. *Machine learning*, 49:161–178, 2002.

James M Ortega and Werner C Rheinboldt. *Iterative solution of nonlinear equations in several variables*. SIAM, 2000.

Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J. Andrew Bagnell, Pieter Abbeel, and Jan Peters. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1–2): 1–179, 2018a. ISSN 1935-8261. doi: 10.1561/2300000053. URL http://dx.doi.org/10.1561/2300000053.

Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2): 1–179, 2018b.

Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients. *Neural Networks*, 21(4):682–697, 2008. URL http://dblp.uni-trier.de/db/journals/nn/nn21.html#PetersS08.

Matteo Pirotta, Marcello Restelli, Alessio Pecorino, and Daniele Calandriello. Safe policy iteration. In *International conference on machine learning*, pp. 307–315. PMLR, 2013.

Dean A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In D. Touretzky (ed.), *Advances in Neural Information Processing Systems*, volume 1. Morgan-Kaufmann, 1988. URL https://proceedings.neurips.cc/paper_files/paper/1988/file/812b4ba287f5ee0bc9d43bbf5bbe87fb-Paper.pdf.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., USA, 1st edition, 1994. ISBN 0471619779.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

Murray Rosenblatt. Remarks on Some Nonparametric Estimates of a Density Function. *The Annals of Mathematical Statistics*, 27(3):832 – 837, 1956. doi: 10.1214/aoms/1177728190. URL https://doi.org/10.1214/aoms/1177728190.

Stéphane Ross and J. Andrew Bagnell. Reinforcement and imitation learning via interactive no-regret learning. *CoRR*, abs/1406.5979, 2014. URL http://arxiv.org/abs/1406.5979.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/ross11a.html.

Juan C Santamaria, Richard S Sutton, and Ashwin Ram. Experiments with reinforcement learning in problems with continuous state and action spaces. *Adaptive behavior*, 6(2):163–217, 1997.

Stefan Schaal. Learning from demonstration. In M.C. Mozer, M. Jordan, and T. Petsche (eds.), *Advances in Neural Information Processing Systems*, volume 9. MIT Press, 1996. URL https://proceedings.neurips.cc/paper_files/paper/1996/file/68d13cf26c4b4f4f932e3eff990093ba-Paper.pdf.

John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In Francis R. Bach and David M. Blei (eds.), *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 1889–1897. JMLR.org, 2015. URL http://dblp.uni-trier.de/db/conf/icml/icml2015.html#SchulmanLAJM15.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://dblp.uni-trier.de/db/journals/corr/corr1707.html#SchulmanWDRK17.

Devavrat Shah and Qiaomin Xie. Q-learning with nearest neighbors. *Advances in Neural Information Processing Systems*, 31, 2018.

Dana Sharon and Michiel van de Panne. Synthesis of controllers for stylized planar bipedal walking. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 2387–2392. IEEE, 2005.

Junhong Shen and Lin F Yang. Theoretically principled deep rl acceleration via nearest neighbor function approximation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9558–9566, 2021.

Ying Sun, Prabhu Babu, and Daniel P Palomar. Majorization-minimization algorithms in signal processing, communications, and machine learning. *IEEE Transactions on Signal Processing*, 65 (3):794–816, 2016.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

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.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020. ISSN 2665-9638. doi: https://doi.org/10.1016/j.simpa.2020.100022. URL https://www.sciencedirect.com/science/article/pii/S2665963820300099.

Hado Van Hasselt and Marco A Wiering. Reinforcement learning in continuous action spaces. In *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, pp. 272–279. IEEE, 2007.

JAEE Van Nunen. A set of successive approximation methods for discounted markovian decision problems. *Zeitschrift fuer operations research*, 20:203–208, 1976.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

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

Ronald J Williams and Leemon C Baird. Tight performance bounds on greedy policies based on imperfect value functions. Technical report, Tech. rep. NU-CCS-93-14, Northeastern University, College of Computer . . . , 1993.

Supplementary Materials

The following content was not necessarily subject to peer review.

A Proofs

A.1 Proof of Proposition 2.3

Statement: If $Q^\star$ is L-Lipschitz then $V^\star$ is L-Lipschitz:

$$|V^\star(s) - V^\star(s')| \leq L\|s - s'\| \quad \forall s, s' \in \mathcal{S}.$$

Proof.

$$\begin{aligned}|V(s) - V(s')| &= |\max_a Q(s,a) - \max_{a'} Q(s',a')| \\ &\leq \max_a |Q(s,a) - Q(s',a)| \\ &\leq L_q\|s - s'\|\end{aligned}$$

where the first inequality follows from the well-known inequality:

$$|\max_x f(x) - \max_x g(x)| \leq \max_x |f(x) - g(x)|,$$

for functions $f, g : \mathcal{X} \to \mathbb{R}$ □

A.2 Proof of Proposition 2.4

Statement: If the transition map f and rewards r are Lipschitz, i.e.:

$$\begin{aligned}\|f(s,a) - f(s',a')\| &\leq L_f\left(\|s - s'\| + \|a - a'\|\right) \\ |r(s,a) - r(s',a')| &\leq L_r\left(\|s - s'\| + \|a - a'\|\right)\end{aligned}$$

for positive scalars L_f, L_r, and the discount factor satisfies $\gamma L_f < 1$, then $Q^\star$ and $V^\star$ are L-lipschitz with $L \leq \frac{L_r}{1-\gamma L_f}$.

Proof. Since the transitions are deterministic, we can define the open-loop q-function:

$$q(s, \mathbf{a}) \triangleq \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)$$

where $\mathbf{a} = [a_0, a_1, \ldots]$ and s_t are the states under that action sequence, from $s_0 = s$. We will show the following two inequalities:

$$|q(s_0, \mathbf{a}) - q(s_0', \mathbf{a})| \leq L\|s_0 - s_0'\| \tag{6}$$

$$|\max_{\mathbf{a}} q(s_0, \mathbf{a}) - \max_{\mathbf{a}'} q(s_0', \mathbf{a}')| \leq L\|s_0 - s_0'\| \tag{7}$$

Again, since transitions are deterministic, for a fixed s_0 the optimal action sequence $a_0 = \pi^\star(s_0), a_1 = \pi^\star(f(s_0, \pi^\star(s_0))), \ldots$ is unique and well-defined. In that way, notice showing (7) is equivalent to showing $V^\star$ is L-Lipschitz.

Let $s_k := \phi(k, s_0, \mathbf{a}_{|k})$ be the solution at time k from s_0 under control law $\mathbf{a}_{|k} = [a_0, \ldots, a_{k-1}]$, and $s_k' := \phi(k, s_0', \mathbf{a}_{|k}')$ be defined similarly. Our bread-and-butter for all the proofs will come from the following inequality, which we show by induction:

$$\|s_k - s_k'\| \leq L_f^k\|s_0 - s_0'\| + \sum_{\ell=0}^{k-1} L_f^{k-1-\ell}\|a_\ell - a_\ell'\| \quad \forall k \geq 0\,. \tag{8}$$

The base case $k = 0$ holds trivially. Assume it holds for time $k-1$ (IH). We then have:

$$\begin{aligned}\|s_k - s'_k\| &= \left\|f(s_{k-1}, a_{k-1}) - f(s'_{k-1}, a'_{k-1})\right\| && (9)\\ &\le L_f \left(\|s_{k-1} - s'_{k-1}\| + \|a_{k-1} - a'_{k-1}\|\right) && (10)\\ &\overset{(IH)}{\le} L_f \left(L_f^{k-1}\|s_0 - s'_0\| + \sum_{\ell=0}^{k-2} L_f^{k-2-\ell}\|a_\ell - a'_\ell\| + \|a_{k-1} - a'_{k-1}\|\right) && (11)\\ &= L_f^k\|s_0 - s'_0\| + \sum_{\ell=0}^{k-1} L_f^{k-1-\ell}\|a_\ell - a'_\ell\| \,. && (12)\end{aligned}$$

To show (6), note that under the same control laws we have, by (8):

$$\|s_k - s'_k\| \le L_f^k\|s_0 - s'_0\| \implies |r(s_k, a_k) - r(s'_k, a_k)| \le L_r L_f^k \|s_0 - s'_0\| \implies \tag{13}$$

$$|q(s_0, \mathbf{a}) - q(s'_0, \mathbf{a})| \le \sum_{k=0}^{\infty} \gamma^k L_r L_f^k \|s_0 - s'_0\| = L\|s_0 - s'_0\|. \tag{14}$$

What remains is to show (7):

$$\begin{aligned}|\max_{\mathbf{a}} q(s_0, \mathbf{a}) - \max_{\mathbf{a}'} q(s'_0, \mathbf{a}')| &\le \max_{\mathbf{a}} |q(s_0, \mathbf{a}) - q(s'_0, \mathbf{a})| && (15)\\ &\le \max_{\mathbf{a}} L\|s_0 - s'_0\| = L\|s_0 - s'_0\| && (16)\end{aligned}$$

where the first inequality follows from the following lemma:

$$|\max_x f(x) - \max_x g(x)| \le \max_x |f(x) - g(x)|$$

□

A.3 Proof of Theorem 3.4

Statement: Let $\mathcal{D}$ be a consistent dataset and π as defined in Definition 3.1. Then:

$$V_{\text{lb}}(s) \le V^\pi(s) \le V^\star(s) \quad \forall s.$$

Proof. To show $V_{\text{lb}}(s) \le V^\pi(s)$ we will make use of the following lemma:

Lemma A.1 ((Bertsekas, 2019)). *If there exists $V : \mathcal{S} \to \mathbb{R}$ such that $V(s) \le r(s, \pi(s)) + \gamma V(f(s, \pi(s)))$ $\forall s \in \mathcal{S}$, then $V(s) \le V^\pi$.*

We will show V_{lb} satisfies the inequality in the lemma above. Fix an arbitrary s. Recall:

$$\begin{aligned}V_{\text{lb}}(s) &= \max_{1\le i\le|\mathcal{D}|} \{Q_i - L\|s - s_i\|\},\\ Q_{\text{lb}}(s, a) &= \max_{1\le i\le|\mathcal{D}|} \{Q_i - L(\|s - s_i\| + \|a - a_i\|)\},\end{aligned}$$

where for each i we have $Q_i = Q^\star(s_i, a_i)$, $a_i = \pi^\star(s_i)$.

We want to show:

$$V_{\text{lb}}(s) \le r(s, \pi(s)) + \gamma V_{\text{lb}}\big(f(s, \pi(s))\big) \quad \forall s \in \mathcal{S},$$

or, equivalently,

$$\mathcal{T}^\pi V_{\text{lb}}(s) - V_{\text{lb}}(s) \ge 0, \tag{17}$$

where we use the short-hand $\mathcal{T}^\pi V_{\text{lb}}(s) = r(s, \pi(s)) + \gamma V_{\text{lb}}\big(f(s, \pi(s))\big)$ for the standard Bellman operator (Bertsekas, 2012).

Fix a state s. Our policy π acts greedily with respect to $Q_{\text{lb}}(s,a)$. With some abuse of notation, let i be the corresponding maximizer of Q_{lb} for that given s. This means a tuple $(s_i, a_i \equiv \pi^\star(s_i))$ gives the largest value for the left hand side.

Starting from (17):

$$\begin{aligned}
\mathcal{T}^\pi V_{\text{lb}}(s) - V_{\text{lb}}(s) &= r(s,\pi(s)) + \gamma V_{\text{lb}}\big(f(s,\pi(s))\big) - V_{\text{lb}}(s) \\
&= r(s,a_i) + \gamma V_{\text{lb}}\big(f(s,a_i)\big) - V_{\text{lb}}(s) \\
&= r(s,a_i) + \gamma V_{\text{lb}}\big(f(s,a_i)\big) - Q_i + L\|s-s_i\| \\
&= Q^\star(s,a_i) - \gamma V^\star\big(f(s,a_i)\big) + \gamma V_{\text{lb}}(f(s,a_i)) - Q_i + L\|s-s_i\| \\
&\geq \underbrace{Q^\star(s_i,a_i)}_{Q_i} - L\|s-s_i\| - \gamma V^\star\big(f(s,a_i)\big) + \gamma V_{\text{lb}}(f(s,a_i)) - Q_i + L\|s-s_i\| \\
&= \gamma V_{\text{lb}}(f(s,a_i)) - \gamma V^\star(f(s,a_i)) \geq 0 \iff \\
&\quad V_{\text{lb}}\big(f(s,a_i)\big) \geq V^\star\big(f(s,a_i)\big) \implies V_{\text{lb}}(s') = V^\star(s').
\end{aligned}$$

Ergo the theorem is true as long as $V_{\text{lb}}(s') = V^\star(s')$ for every successor state $s' = f(s_i,a_i)$ for tuples (s_i,a_i) belonging to the dataset. But this is true, because by Assumption 2.5 our data comes from *expert trajectories*. Therefore V_{lb} satisfies the condition of the lemma, and then $V_{\text{lb}}(s) \leq V^\pi(s)$. □

A.4 Proof of Theorem 3.5

Statement: Let $\mathcal{D}$, $\mathcal{D}'$ be consistent datasets with $\mathcal{D} \subset \mathcal{D}'$. Let V_{lb} and V'_{lb} be the lower bounds constructed with $\mathcal{D}$ and $\mathcal{D}'$ respectively. Then the following **non-deterioration** condition holds:

- $V_{\text{lb}}(s) \leq V'_{\text{lb}}(s), \forall s \in \mathcal{S}$, and
- $V^\pi(s) \leq V^{\pi'}(s), \forall s \in \Pi_{\mathcal{S}}[\mathcal{D}' \backslash \mathcal{D}]$,

where $\Pi_{\mathcal{S}}[\mathcal{D}] \triangleq \{s_i : \exists a_i, Q_i \text{ such that } (s_i,a_i,Q_i) \in \mathcal{D}\}$. Furthermore, if there exists $s' \in \Pi_{\mathcal{S}}[\mathcal{D}' \backslash \mathcal{D}]$ and a neighborhood $N(s')$ such that $\sup_{s\in N(s')} V^\pi(s) < V^\star(s')$, then **strict improvement** exists in $N(s')$:

- $V_{\text{lb}}(s) < V'_{\text{lb}}(s), \forall s \in N(s')$, and
- $V^\pi(s) < V^{\pi'}(s), \forall s \in N(s')$.

Proof. We start with the *non-deterioration* conditions. Note $\mathcal{D} \subset \mathcal{D}' \implies |\mathcal{D}| \leq |\mathcal{D}'|$ and therefore:

$$\forall s \in \mathcal{S} \;\; V_{\text{lb}}(s) = \max_{1\leq i\leq|\mathcal{D}|} \{Q_i - L\|s-s_i\|\} \leq \max_{1\leq i\leq|\mathcal{D}'|} \{Q_i - L\|s-s_i\|\},$$

proving the first point. For the second one, note that $\forall s \in \Pi_{\mathcal{S}}[\mathcal{D}' \backslash \mathcal{D}]$ we have $V^{\pi'}(s) = V^\star(s) \geq V^\pi(s)$.

We now show the *strict-improvement* conditions. Assuming: $\sup_{s\in\mathcal{B}(s')} V^\pi(s) < V^\star(s')$.

We will show $V'_{\text{lb}}(s) > V^\pi(s)$ on some neighborhood $N(s')$. Note that adding the triplet (s', a', Q') yields:

$$V'_{\text{lb}}(s) \geq \underbrace{Q'}_{=V^\star(s')} - L\|s-s'\| > V^\pi(s) \iff \frac{Q' - V^\pi(s)}{L} > \|s-s'\|$$

Note $Q' - V^\pi(s) \geq Q' - \sup_{s\in\mathcal{B}(s')} V^\pi(s) =: \Delta V$. Then, if $|s-s'\| < \frac{\Delta V}{L}$ and $s \in \mathcal{B}(s')$, we have $V'_{\text{lb}}(s) > V^\pi(s)$, as desired. Invoking Theorem 3.4, we know $V^{\pi'} \geq V'_{\text{lb}} \implies$

$$V^{\pi'}(s) > V^\pi(s) \quad \forall s \in N(s') \triangleq \left\{s \in \mathcal{B}(s') : \|s-s'\| \leq \tfrac{\Delta V}{L}\right\}$$

$$\eta \triangleq V^\star(s') - \sup_{s \in N(s')} V^\pi(s) > 0.$$

By the Lipschitz property of $V^\star$, we know

$$V^\star(s) \geq V^\star(s') - L\|s - s'\| \quad \forall s \in \mathcal{S}.$$

Define $\mathcal{B}(s') = \left\{s \in \mathcal{S} : \|s - s'\| \leq \frac{0.9\eta}{L}\right\}$. Then:

$$\forall s \in \mathcal{B}(s') \quad V^\star(s) \geq V^\star(s') - L\|s - s'\| = V'_{\mathrm{lb}}(s) > V^\pi(s).$$

Since the new policy π' acts greedily with respect to the lower bound, we have

$$V^{\pi'}(s) \geq V'_{\mathrm{lb}}(s) > V^\pi(s) \quad \forall s \in \mathcal{B}(s')$$

$\square$

A.5 Proof of Theorem 3.7

Statement: If for all $s \in \mathcal{S}$ there exists $s_i \in \Pi_{\mathcal{S}}[\mathcal{D}]$ such that:

$$\|s - s_i\| \leq \frac{\varepsilon}{2L},$$

then π is $\varepsilon-$suboptimal.

Proof. By the fact that $V_{\mathrm{lb}}(s) \leq V^\pi(s)$, we have:

$$Q_i - L\|s - s_i\| \leq V^\pi(s).$$

On the other hand, by the Lipschitz assumption on $V^\star$,

$$V^\star(s) \leq \overbrace{V^\star(s_i)}^{\equiv Q_i} + L\|s - s_i\|$$

We substract these two inequalities and enforce the ε-suboptimality:

$$V^\star(s) - V^\pi(s) \leq Q_i + L\|s - s_i\| - V^\pi(s) \leq 2L\|s - s_i\| \leq \varepsilon \implies \|s - s_i\| \leq \frac{\varepsilon}{2L}$$

$\square$

A.6 Proof of Theorem 4.1

Statement: Let Δ_e be defineds as in Algorithm 1 for each episode e. Let $\mathcal{S}_0 \triangleq \mathrm{supp}(\rho)$.

i) If for the last n episodes no new data has been collected, then with probability at least $1 - \delta$, we have $\mathbb{P}_{s \sim \rho}[V^\star(s) - V^\pi(s) \leq \varepsilon] \geq p$, provided:

$$n \geq \frac{1}{1-p} \log \frac{1}{\delta}$$

ii) Suppose $\bar{\Delta}_n \triangleq \frac{1}{n}\sum_{e=1}^{n} \Delta_e \leq \frac{\varepsilon}{2L}$. Then with probability at least $1 - \delta$ we have $\mathbb{E}_{s \sim \rho}[V^\star(s) - V^\pi(s)] \leq \varepsilon$, provided $\bar{\Delta}_n \leq \frac{\varepsilon}{2L}$ and

$$n \geq \frac{2L^2 \operatorname{diam}(\mathcal{S}_0)}{(\varepsilon - 2L\bar{\Delta}_n)^2} \log \frac{1}{\delta}\ .$$

Proof. *i)* This follows from a standard result in PAC learnability (Kearns & Vazirani, 1994). Let the random variable W be defined over $\mathcal{S}_0$ such that $W(s) \triangleq \mathbb{1}\{V^\star(s) - V^\pi(s) > \varepsilon\} \sim \text{Bernoulli}(q)$. Assume $q \geq 1 - p$.

Let Δ_i be the distance from the initial state in episode i to its "closest" datapoint, in the sense of (1) (see Algorithm 1). If no new data has been collected for the last n episodes, this means:

$$\forall 1 \leq i \leq n \quad \Delta_i \leq \frac{\varepsilon}{2L} \implies V^\star(s_i) - V^\pi(s_i) \leq \varepsilon$$

Then:

$$\mathbb{P}\left[\bigcap_{i=1}^{n}\left\{\Delta_i \leq \frac{\varepsilon}{2L}\right\}\right] \leq \mathbb{P}\left[\bigcap_{i=1}^{n}\{W_i = 0\}\right] = (1-q)^n \leq p^n \leq e^{-(1-p)n} \leq \delta \implies n \geq \frac{1}{1-p}\log\frac{1}{\delta}\,.$$

where in the second inequality we use the approximation $(1-x) \leq e^{-x}$ for all $x \in [0, 1]$.

ii) We consider the last n rounds of the algorithm, and define:

$$V_e \triangleq V^\star(s_e) - V^\pi(s_e) \quad e = 1 \ldots n$$

where $s_e \sim \rho$ was the state sampled at episode e. Clearly

$$\mathbb{E}[V_e] = \mathbb{E}_{s\sim\rho}[V^\star(s) - V^\pi(s)]$$

Notice, by Theorem 3.7 that since:

$$2L\Delta_e \leq \varepsilon \implies V_e \leq \varepsilon,$$

we have the event inclusion

$$\{2L\Delta_e\varepsilon\} \supset \{V_e \leq \varepsilon\}\,.$$

Furthermore, Δ_e are bounded almost surely:

$$0 \leq \Delta_e \leq \sup_{s,s'\in\mathcal{S}} \|s - s'\| = \operatorname{diam}(\mathcal{S}_0),$$

where $\mathcal{S}_0 = \operatorname{supp}(\rho)$. Applying Hoeffding's bound (Thm. 2.2.6 in (Vershynin, 2018)):

$$\mathbb{P}\left[\bar{V}_n - \mathbb{E}_{s\sim\rho}[V^\star(s) - V^\pi(s)] \leq -t\right] \leq \mathbb{P}\left[\bar{\Delta}_n - \mathbb{E}\Delta \leq -t\right] \leq \exp\left(\frac{-2t^2 n}{4L^2\operatorname{diam}^2(\mathcal{S}_0)}\right) \leq \delta \implies$$

$$n \geq \frac{2L^2\operatorname{diam}^2(\mathcal{S}_0)}{t^2}\log\frac{1}{\delta}$$

Choosing $t = \varepsilon - 2L\bar{\Delta}_n$ (and $0 < t$ by assumption) gives the desired result.

□

B Policy evaluation/improvement with suboptimal data

What happens if the demonstrations come from a suboptimal policy? We provide theoretical insight by extending theorems 3.4–3.5 and with numerical simulations that serve as proof of concept to our approach.

Theorem B.1 (Policy improvement with suboptimal data)**.** *Let $\mathcal{D} = \{(s_i, a_i, Q_i)\}_i$ be a dataset containing trajectories collected by a policy $\tilde{\pi}$, i.e. $a_i = \tilde{\pi}(s_i), Q_i = Q^{\tilde{\pi}}(s_i, a_i)$. Assume $Q^{\tilde{\pi}}$ is L-Lipschitz. Define the lower bounds $\tilde{Q}_{\text{lb}}$ and $\tilde{V}_{\text{lb}}$ analogously to* (1) *and* (2)*.*

Let $\pi(s) = \arg\max_{a \in \mathcal{A}} \tilde{Q}_{\text{lb}}(s, a)$. Then:

i) *(Evaluation) $\tilde{V}_{\text{lb}} \leq V^{\pi} \leq V^{\star}(s) \quad \forall s$.*

ii) *(Improvement) Assume $V^{\pi}(s) \leq V^{\tilde{\pi}}(s)\ \forall s$. Then, if $\mathcal{D}' \supset \mathcal{D} \implies V^{\pi}(s) \leq V^{\pi'}(s)\ \forall s \in \Pi_{\mathcal{S}}\left[\mathcal{D}' \backslash \mathcal{D}\right]$.*

Experiments To support the discussion in Section 3, we used the `Pendulum Swing-Up` environment from the `DeepMind Control Suite` to investigate the case where the dataset is generated by a suboptimal policy.

The environment is a nonlinear control problem where the goal is to swing up and stabilize a freely hanging pendulum. The state consists of the pendulum's angular position and velocity, $\dim(\mathcal{S}) = 2$, and the action space is a single torque input, $\dim(\mathcal{A}) = 1$.

To generate suboptimal trajectories, we trained an agent using `Proximal Policy Optimization (PPO)` (Schulman et al., 2017) with `Stable-Baselines3` (Raffin et al., 2021). The expert was trained for 1 million timesteps with a discount factor of $\gamma = 0.99$ and a batch size of 256.

For evaluation, we set the suboptimality gap to $\varepsilon = 130$ and ran the environment with different seeds of the evaluation space and evaluated $N = 50$ rollouts per episode. The `NPP` algorithm used a Lipschitz constant of $L = 4300$ and a horizon of $H = 1000$. As shown in figure 7, the rightmost plot, prior to the 320$^{\text{th}}$ episode, the surrogate gap is below ε for consecutive episodes.

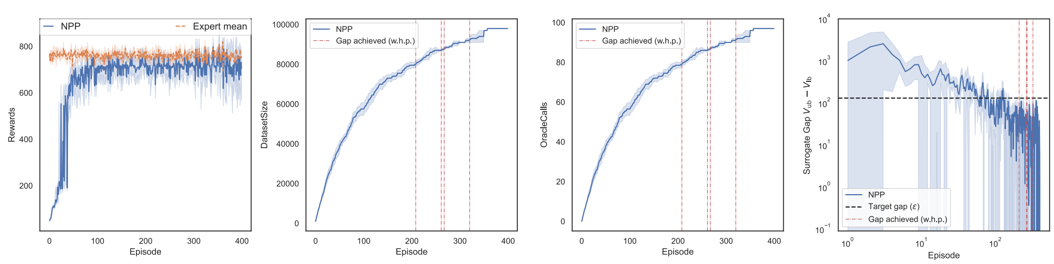

Figure 7: Training curves for `Pendulum Swing-Up` with target suboptimality $\varepsilon = 130$, with results averaged over 4 seeds. *Left*: Episodic return of policy π (in blue) and expert (in orange) at different stages of training. $N = 50$ rollouts are performed at each point; solid line corresponds to the median and shaded area to a 95% confidence interval. *Middle-left*: size of the dataset. *Middle-right*: calls to the `TrajectoryOptimizer` oracle (notice calls are made on approximately one third of the episodes). *Right*: surrogate gap $V_{\text{ub}} - V_{\text{lb}}$ for the initial states. Purple dashed lines correspond to the hitting times (one per seed) for reaching the target suboptimality gap.

C Environment testing

We ran 1000 episodes of the optimal controller for both lqr environments, in order to come up with an estimate of the Lipschitz constant for the value function under the optimal policy. The results are on Figures 8 and 9.

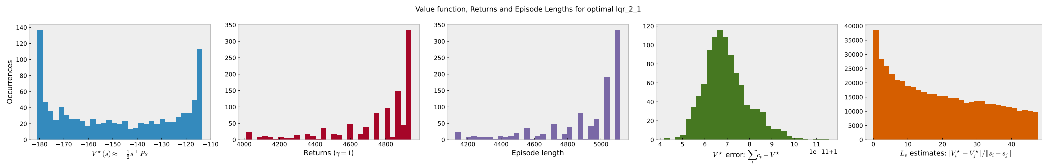

Figure 8: Statistics for `lqr_2_1`. The right-most histogram justifies the choice of $L \approx 50$.

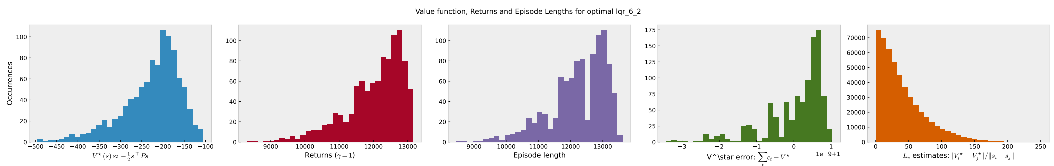

Figure 9: Statistics for `lqr_6_2`. The right-most histogram justifies the choice of $L \approx 200$.

D Additional experimental results

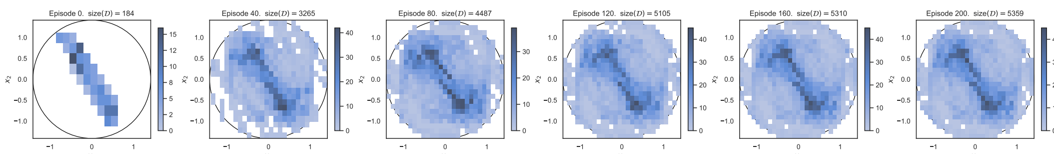

Figure 10: Dataset collected by the policy at different stages of training on environment `lqr_2_1`.

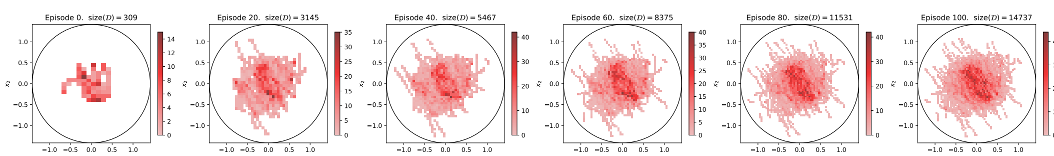

Figure 11: Dataset collected by the policy at different stages of training on environment `lqr_6_2`.