Safety Augmented Value Estimation from Demonstrations (SAVED): Safe Deep Model-Based RL for Sparse Cost Robotic Tasks

Brijen Thananjeyan*, Ashwin Balakrishna*, Ugo Rosolia, Felix Li, Rowan McAllister,
Joseph E. Gonzalez, Sergey Levine, Francesco Borrelli, Ken Goldberg

Abstract— **Reinforcement learning (RL) for robotics is challenging due to the difficulty in hand-engineering a dense cost function, which can lead to unintended behavior, and dynamical uncertainty, which makes exploration and constraint satisfaction challenging. We address these issues with a new model-based reinforcement learning algorithm, Safety Augmented Value Estimation from Demonstrations (SAVED), which uses supervision that only identifies task completion and a modest set of suboptimal demonstrations to constrain exploration and learn efficiently while handling complex constraints. We derive iterative improvement guarantees for SAVED under known stochastic nonlinear systems. We then compare SAVED with 3 state-of-the-art model-based and model-free RL algorithms on 6 standard simulation benchmarks involving navigation and manipulation and a knot-tying task on the da Vinci surgical robot. Results suggest that SAVED outperforms prior methods in terms of success rate, constraint satisfaction, and sample efficiency, making it feasible to safely learn maneuvers directly on a real robot in less than an hour. For tasks on the robot, baselines succeed less than 5% of the time while SAVED has a success rate of over 75% in the first 50 training iterations. Code and supplementary material is available at `https://tinyurl.com/saved-rl`.**

I. Introduction

To use RL in the real world, algorithms need to be efficient, easy to use, and safe, motivating methods which are reliable even with significant dynamical uncertainty. Deep model-based reinforcement learning (deep MBRL) is of significant interest because of its sample efficiency advantages over model-free methods in a variety of tasks, such as assembly, locomotion, and manipulation [7–9, 15, 16, 21, 25]. However, past work in deep MBRL typically requires dense hand-engineered cost functions, which are hard to design and can lead to unintended behavior [2]. It would be easier to simply specify task completion in the cost function, but this setting is challenging due to the lack of expressive supervision. This motivates using demonstrations, which allow the user to roughly specify desired behavior without extensive engineering effort. Furthermore, in many robotic tasks, specifically in domains such as surgery, safe exploration is critical to ensure that the robot does not damage itself or cause harm to its surroundings. To enable this, deep MBRL algorithms also need the ability to satisfy complex constraints.

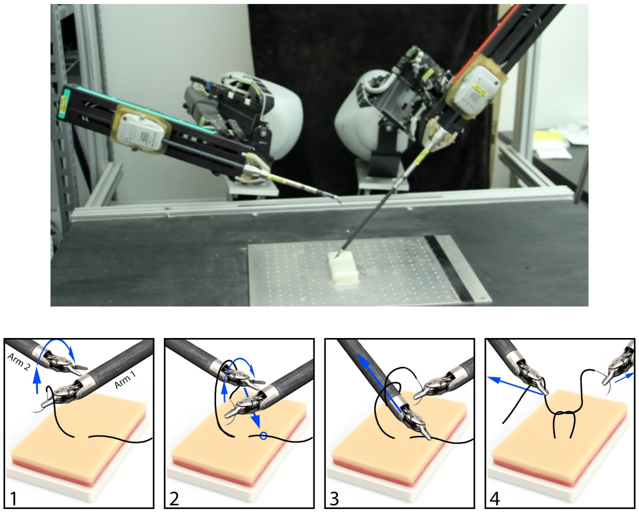

Fig. 1: SAVED is able to safely learn maneuvers on the da Vinci surgical robot, which is difficult to precisely control [29]. We demonstrate that SAVED is able to optimize inefficient human demonstrations of a surgical knot-tying task, substantially improving on demonstration performance with just 15 training iterations.

We develop a method to efficiently use deep MBRL in dynamically uncertain environments with both sparse costs and complex constraints. We address the difficulty of hand-engineering cost functions by using a small number of suboptimal demonstrations to provide a signal about delayed costs in sparse cost environments, which is updated based on agent experience. Then, to enable stable policy improvement and constraint satisfaction, we impose two probabilistic constraints to (1) constrain exploration by ensuring that the agent can plan back to regions in which it is confident in task completion and (2) leverage uncertainty estimates in the learned dynamics to implement chance constraints [23] during learning. The probabilistic implementation of constraints makes this approach broadly applicable, since it can handle settings with significant dynamical uncertainty, where enforcing constraints exactly is difficult.

We introduce a new algorithm motivated by deep model predictive control (MPC) and robust control, Safety Augmented Value Estimation from Demonstrations (SAVED), which enables efficient learning for sparse cost tasks given a small number of suboptimal demonstrations while satisfying provided constraints. We specifically consider tasks with a tight start state distribution and fixed, known goal set, and only use supervision that indicates task completion. We then show that under certain regularity assumptions and given known stochastic nonlinear dynamics, SAVED has guaranteed iterative improvement in expected performance, extending prior analysis of similar methods for known stochastic linear dynamics [27, 28]. The contributions of this work are (1) a novel method for constrained exploration driven by confidence

* Equal contribution
[1]University of California, Berkeley

in task completion, (2) a technique for leveraging model uncertainty to probabilistically enforce complex constraints, enabling obstacle avoidance or optimizing demonstration trajectories while maintaining desired properties, (3) analysis of SAVED which provides iterative improvement guarantees in expected performance for known stochastic nonlinear systems, and (4) experimental evaluation against 3 state-of-the-art model-free and model-based RL baselines on 8 different environments, including simulated experiments and physical maneuvers on the da Vinci surgical robot. Results suggest that SAVED achieves superior sample efficiency, success rate, and constraint satisfaction rate across all domains considered and can be applied efficiently and safely for learning directly on a real robot.

II. Related work

There is significant interest in deep MBRL [7–9, 15, 18, 21] due to the improvements in sample efficiency when planning over learned dynamics compared to model-free methods for continuous control [10, 12]. However, most prior deep MBRL algorithms use hand-engineered dense cost functions to guide exploration, which we avoid by using demonstrations to provide signal about delayed costs. Demonstrations have been leveraged to accelerate learning for a variety of model-free RL algorithms, such as Deep Q Learning [13] and DDPG [22, 33], but model-free methods are typically less sample efficient and cannot anticipate constraint violations since the policy is reactive [31]. Fu *et al.* [9] use a neural network prior from previous tasks and online adaptation to a new task using iLQR and a dense cost, distinct from the task completion based costs we consider. Finally, Brown *et al.* [6] use inverse RL to significantly outperform suboptimal demonstrations, but do not explicitly optimize for constraint satisfaction or consistent task completion during learning.

In iterative learning control (ILC), the controller tracks a predefined reference trajectory and data from each iteration is used to improve closed-loop performance [5]. Rosolia *et al.* [26–28] provide a reference-free algorithm to iteratively improve the performance of an initial trajectory by using a safe set and terminal cost to ensure recursive feasibility, stability, and local optimality given a known, deterministic nonlinear system or stochastic linear system under certain regularity assumptions. We extend this analysis, and show that given task completion based costs, similar guarantees hold for stochastic nonlinear systems with bounded disturbances satisfying similar assumptions. Furthermore, in contrast to Rosolia *et al.* [26–28], SAVED is designed for settings with completely unknown dynamics and continuous state spaces, which requires function approximation to estimate a dynamics model, value function, and safe set. There has also been significant interest in safe RL [11], typically focusing on exploration while satisfying a set of explicit constraints [1, 17, 20], satisfying specific stability criteria [3], or formulating planning via a risk sensitive Markov Decision Process [19, 24]. Distinct from prior work in safe RL and control, SAVED can be successfully applied in settings with both uncertain dynamics and sparse costs by using probabilistic constraints to constrain exploration to feasible regions during learning.

III. Safety Augmented Value Estimation from Demonstrations (SAVED)

This section describes how SAVED uses a set of suboptimal demonstrations to constrain exploration while satisfying user-specified state space constraints. First, we discuss how SAVED learns system dynamics and a value function to guide learning in sparse cost environments. Then, we motivate and discuss the method used to enforce constraints under uncertainty to both ensure task completion during learning and satisfy user-specified state space constraints.

A. Assumptions and Preliminaries

In this work, we consider stochastic, unknown dynamical systems with a cost function that only identifies task completion. We assume that (1) tasks are iterative in nature, and thus have a fixed low-variance start state distribution and fixed, known goal set $\mathcal{G}$. This is common in a variety of repetitive tasks, such as assembly, surgical knot-tying, and suturing. Additionally, we assume that (2) a modest set of suboptimal but successful demos are available, for example from imprecise human teleoperation or a hand-tuned PID controller. This enables rough specification of desired behavior without having to design a dense cost function.

Here we outline the framework for MBRL using a standard Markov Decision Process formulation. A finite-horizon Markov Decision Process (MDP) is a tuple $(\mathcal{X}, \mathcal{U}, P(\cdot,\cdot), T, C(\cdot,\cdot))$ where $\mathcal{X}$ is the feasible state space and $\mathcal{U}$ is the action space. The stochastic dynamics model P maps a state and action to a probability distribution over states, T is the task horizon, and C is the cost function. A stochastic control policy π maps an input state to a distribution over $\mathcal{U}$. We assume that the cost function only identifies task completion: $C(x,u) = \mathbb{1}_{\mathcal{G}^C}(x)$, where $\mathcal{G} \subset \mathcal{X}$ defines a goal set in the state space and $\mathcal{G}^C$ is its complement. We define task success by convergence to $\mathcal{G}$ at the end of the task horizon without violating constraints.

B. Algorithm Overview

1) Deep Model Predictive Control: SAVED optimizes agent trajectories by using MPC to optimize costs over a sequence of actions at each state. However, when using MPC, since the current control is computed by solving a finite-horizon approximation to the infinite-horizon control problem, agents may take shortsighted actions which may make it impossible to complete the task, such as planning the trajectory of a race car over a short horizon without considering an upcoming curve [4]. Thus, to guide exploration in temporally-extended tasks, we solve the problem in equation III-B.1a, which includes a learned value function in the objective.

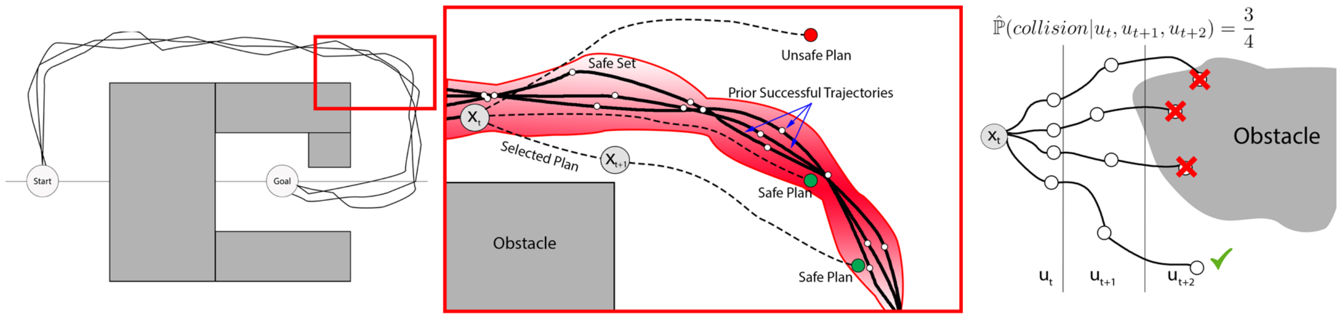

Fig. 2: **Task Completion Driven Exploration (left):** A density model is used to represent the region in state space where the agent has high confidence in task completion; trajectory samples over the learned dynamics that do not have sufficient density at the end of the planning horizon are discarded. The agent may explore outside the safe set as long as a plan exists to guide the agent back to the safe set from the current state; **Chance Constraint Enforcement (right):** Implemented by sampling imagined rollouts over the learned dynamics for the same sequence of actions multiple times and estimating the probability of constraint violation by the percentage of rollouts that violate a constraint.

$$u^*_{t:t+H-1} = \underset{u_{t:t+H-1}\in\mathcal{U}^H}{\operatorname{argmin}} \; \mathbb{E}_{x_{t:t+H}}\left[\sum_{i=0}^{H-1} C(x_{t+i},u_{t+i}) + V^{\pi}_{\phi}(x_{t+H})\right] \tag{III-B.1a}$$

$$\text{s.t. } x_{t+i+1} \sim f_\theta(x_{t+i},u_{t+i}) \;\forall i \in \{0,\dots,H-1\} \tag{III-B.1b}$$

$$\rho_\alpha(x_{t+H}) > \delta, \mathbb{P}\left(x_{t:t+H} \in \mathcal{X}^{H+1}\right) \geq \beta \tag{III-B.1c}$$

Note that $\mathcal{U}^H$ refers to the set of H length action sequences while $\mathcal{X}^{H+1}$ refers to the set of $H+1$ length state sequences. This corresponds to the standard objective in MPC with an appended value function V^π, which provides a terminal cost estimate for the current policy at the end of the planning horizon. While prior work in deep MBRL [7, 21] has primarily focused only on planning over learned dynamics, we introduce a learned value function, which is initialized from demonstrations to provide initial signal, to guide exploration even in sparse cost settings. The learned dynamics model f_θ and value function V^π_ϕ are each represented with a probabilistic ensemble of 5 neural networks, as is used to represent system dynamics in Chua *et al.* [7]. These functions are initialized from demonstrations and updated on each training iteration, and collectively define the current policy $\pi_{\theta,\phi}$. See supplementary material for further details on how these networks are trained.

2) Probabilistic Constraints: The core novelties of SAVED are the additional probabilistic constraints in III-B.1c to encourage task completion driven exploration and enforce user-specified chance constraints. First, a non-parametric density model ρ is trained on states from prior successful trajectories, including demos. ρ enforces constrained exploration by requiring x_{t+H} to fall in a region with high probability of task completion. This enforces cost-driven constrained exploration, enabling reliable performance even with sparse costs. Second, we require all elements of $x_{t:t+H}$ to fall in the feasible region $\mathcal{X}$ with probability at least β, which enables probabilistic enforcement of state space constraints. In Section III-C, we discuss the methods used for task completion driven exploration and in Section III-D, we discuss how probabilistic constraints are enforced during learning.

Algorithm 1 Safety Augmented Value Estimation from Demonstrations (SAVED)

```
Require: Replay Buffer $\mathcal{R}$; value function $V^\pi_\phi(x)$, dynamics model $\hat{f}_\theta(x'|x,u)$, and safety density model $\rho_\alpha(x)$ all seeded with demos; kernel and chance constraint parameters $\alpha$ and $\beta$.
for $i \in \{1,\dots,N\}$ do
    Sample $x_0$ from start state distribution
    for $t \in \{1,\dots,T-1\}$ do
        Pick $u^*_{t:t+H-1}$ by solving eq. III-B.1 using CEM
        Execute $u^*_t$ and observe $x_{t+1}$
        $\mathcal{R} = \mathcal{R} \cup \{(x_t, u^*_t, C(x_t,u^*_t), x_{t+1})\}$
    end for
    if $x_T \in \mathcal{G}$ then
        Update safety density model $\rho_\alpha$ with $x_{0:T}$
    end if
    Optimize $\theta$ and $\phi$ with $\mathcal{R}$
end for
```

C. Task Completion Driven Exploration

Recent MPC literature [26] motivates constraining exploration to regions in which the agent is confident in task completion, which gives rise to desirable theoretical properties. For a trajectory at iteration k, given by x^k, we define the *sampled safe set* as

$$\mathcal{SS}^j = \bigcup_{k\in\mathcal{M}^j} x^k \tag{III-C.2}$$

where $\mathcal{M}^j = \{k \in [0,j) : \lim_{t\to\infty} x^k_t \in \mathcal{G}\}$ is the set of indices of all successful trajectories before iteration j as in Rosolia *et al.* [26]. Thus, $\mathcal{SS}^j$ contains the states from all iterations before j from which the agent controlled the system to $\mathcal{G}$ and is initialized from demonstrations. Under certain regularity assumptions, if states at the end of the MPC planning horizon are constrained to fall in $\mathcal{SS}^j$, iterative improvement, controller feasibility, and convergence are guaranteed given known stochastic linear dynamics or deterministic nonlinear dynamics [26–28]. In Section IV, we extend these results to show that, under similar assumptions, we can obtain the same guarantees in expectation for stochastic nonlinear systems if task completion based costs are used. The way we constrain exploration in SAVED builds off of this prior work, but we note that unlike Rosolia *et al.* [26–28], SAVED is designed for settings in which dynamics are completely unknown. As illustrated in Figure 2, this constraint allows the agent to generate trajectories that leave the sampled safe set as long as a plan exists to navigate back in, enabling policy improvement. By adding newly successful trajectories to the safe set, the agent is able to further improve its performance.

We develop a method to approximately implement the above constraint with a continuous approximation to $\mathcal{SS}^j$ using non-parametric density estimation, allowing SAVED to scale to more complex settings than prior work using similar cost-driven exploration techniques [26–28]. Since $\mathcal{SS}^j$ is a discrete set, we introduce a new continuous approximation by fitting a density model ρ to $\mathcal{SS}^j$ and constraining $\rho_\alpha(x_{t+H}) > \delta$, where α is a kernel width parameter (constraint III-B.1c). Since the tasks considered in this work have sufficiently low (< 17) state space dimension, kernel density estimation provides a reasonable approximation. We implement a tophat kernel density model using a nearest neighbors classifier with a tuned kernel width α and use $\delta = 0$ for all experiments. Thus, all states within Euclidean distance α from the closest state in $\mathcal{SS}^j$ are considered safe under ρ_α, representing states in which the agent is confident in task completion. As the policy improves, it may forget how to complete the task from very old states in $\mathcal{SS}^j$, so such states are evicted from $\mathcal{SS}^j$ to reflect the current policy when fitting ρ_α. We discuss how these constraints are implemented in Section III-D, with further details in the supplementary material. In future work, we will investigate implicit density estimation techniques to scale to high-dimensional settings.

D. Probabilistic Constraint Enforcement

SAVED leverages uncertainty estimates in the learned dynamics to enforce probabilistic constraints on its trajectories. This allows SAVED to handle complex, user-specified state space constraints to avoid obstacles or maintain certain properties of demonstrations without a user-shaped or time-varying cost function. We do this by sampling sequences of actions from a truncated Gaussian distribution that is iteratively updated using the cross-entropy method (CEM) [7]. Each action sequence is simulated multiple times over the stochastic dynamics model as in [7] and the average return of the simulations is used to score the sequence. However, unlike Chua *et al.* [7], we implement chance constraints by discarding actions sequences if more than $100 \cdot (1-\beta)\%$ of the simulations violate constraints (Constraint III-B.1c), where β is a user-specified tolerance. Note that the β parameter essentially controls the tradeoff between ensuring sufficient exploration to learn the dynamics and satisfying specified constraints. This is illustrated in Figure 2. The task completion constraint (Section III-C) is implemented similarly, with action sequences discarded if any of the simulated rollouts do not terminate in a state with sufficient density under ρ_α.

E. Algorithm Pseudocode

We summarize SAVED in Algorithm 1. The dynamics, value function, and state density model are initialized from suboptimal demonstrations. At each iteration, we sample a start state and then controls are generated by solving equation III-B.1 using the cross-entropy method (CEM) at each timestep. Transitions are collected in a replay buffer to update the dynamics, value function, and safety density model at the end of each iteration. The state density model is only updated if the last trajectory was successful.

IV. Theoretical Analysis of SAVED

In prior work, a *sampled safe set* $\mathcal{SS}^j$ and value function were used to design a controller with feasibility, convergence, and iterative improvement guarantees under certain regularity assumptions [27]. Prior work specifically assumes known stochastic linear dynamics, that the limit of infinite data is used for policy evaluation at each iteration, and that the MPC optimal control problem can be solved robustly or exactly [27, 28]. We extend this by showing that under the same assumptions, if task completion based costs (as defined in Section III-A) are used and $\beta = 1$, then the same guarantees can be shown in expectation for SAVED in closed-loop with stochastic nonlinear systems.

A. Definitions and Assumptions

Consider the stochastic dynamical system at time t of iteration j:

$$x^j_{t+1} = f(x^j_t, u^j_t, w^j_t) \tag{IV-A.1}$$

for state $x \in \mathcal{X}$, input $u \in \mathcal{U}$ and disturbance $w \in \mathcal{W}$. Here $\mathcal{X} \subseteq \mathbb{R}^n$ defines the set of feasible states, $\mathcal{U} \subseteq \mathbb{R}^d$ defines the set of allowed controls, and $\mathcal{G} \subseteq \mathbb{R}^n$ defines the *goal set*. The task is considered to be successfully completed on iteration j if $\lim_{t\to\infty} x^j_t \in \mathcal{G}$. In practice, we use a finite task horizon.

Assumption IV.1. *Known stochastic dynamics with bounded disturbances: The dynamics* (IV-A.1) *are known and the set of disturbances $\mathcal{W}$ is bounded.*

Note that while for analysis we assume known dynamics with bounded disturbances, SAVED is designed in practice for unknown, stochastic dynamical systems.

Definition IV.1. *With the sampled safe set $\mathcal{SS}^j$ defined as in III-C.2, recursively define the value function of π^j (SAVED at iteration j) in closed-loop with (IV-A.1) as:*

$$V^{\pi^j}(x) = \begin{cases} \mathbb{E}_w\left[C(x,\pi^j(x)) + V^{\pi^j}(f(x,\pi^j(x),w))\right] & x \in \mathcal{SS}^j \cap \mathcal{X} \\ +\infty & x \notin \mathcal{SS}^j \cap \mathcal{X} \end{cases} \tag{IV-A.2}$$

In the practical implementation of SAVED, we train a value function approximator using TD-1 error [30] corresponding to the standard Bellman equations.

In the analysis, at each timestep we optimize over the set of causal feedback policies Π, ie. policies which only consider the current and prior states, rather than directly over controls as in the practical implementation of SAVED. SAVED optimizes over controls (constant policies) to maintain efficient re-planning. Furthermore, for analysis we consider robust constraints ($\beta = 1$); note that the value function implicitly constrains terminal states to robustly fall within $\mathcal{SS}^j$.

Specifically, the optimization problem at time t of iteration j is to find $\pi^{*,j}_{t:t+H-1|t}$ (the optimal sequence of policies for the MPC cost conditioned on x^j_t), which is defined as follows:

$$\begin{aligned} \underset{\pi_{t:t+H-1|t} \in \Pi^H}{\operatorname{argmin}} \ & \mathbb{E}_{x^j_{t:t+H|t}} \left[\sum_{i=0}^{H-1} C(x^j_{t+i|t}, \pi_{t+i|t}(x^j_{t+i|t})) + V^{\pi^{j-1}}(x^j_{t+H|t}) \right] \\ \text{s.t. } & x^j_{t+i+1|t} = f(x^j_{t+i|t}, \pi_{t+i|t}(x^j_{t+i}), w_{t+i}) \ \forall i \in \{0, \ldots, H-1\} \\ & x^j_{t+H|t} \in \mathcal{SS}^j, \ \forall w_t \in \mathcal{W} \\ & x^j_{t:t+H|t} \in \mathcal{X}^{H+1}, \ \forall w_t \in \mathcal{W} \end{aligned} \tag{IV-A.3}$$

π^j is the policy (SAVED) at iteration j, where

$$u^j_t = \pi^j(x^j_t) = \pi^{*,j}_{t|t}(x^j_t) \tag{IV-A.4}$$

is the control applied at state x_t. $J^j_{t \to t+H}(x^j_t)$ is defined as the value of IV-A.3. For analysis, we assume that we can solve this problem at each timestep and exactly compute V^{π^j}.

Assumption IV.2. *Exact solution to MPC objective and value function: For analysis, we assume that we can solve* (IV-A.3) *exactly and solve the system of equations defining V^{π^j}.*

Note that in practice SAVED does not require that the MPC objective can be solved exactly or that the value function can be estimated exactly. Instead SAVED uses CEM and function approximation to solve the MPC objective and estimate the value function respectively.

Definition IV.2. *We define the planning cost of the controller at time t of iteration j as:*

$$J^j_{t \to t+H}(x^j_t) = \min_{\pi_{t:t+H-1|t}} \mathbb{E}_{x^j_{t:t+H|t}} \left[\sum_{k=t}^{t+H-1} C(x^j_{k|t}, \pi_{k|t}(x^j_{k|t})) + V^{\pi^{j-1}}(x^j_{t+H|t}) \right] \tag{IV-A.5}$$

$$= \mathbb{E}_{x^j_{t:t+H|t}} \left[\sum_{k=t}^{t+H-1} C(x^j_{k|t}, \pi^{*,j}_{k|t}(x^j_{k|t})) + V^{\pi^{j-1}}(x^j_{t+H|t}) \right] \tag{IV-A.6}$$

where $\pi^{*,j}_{t:t+H-1|t}$ is the minimizer of IV-A.5. Note that this enforces the safe set constraint through the support of $V^{\pi^{j-1}}$. SAVED therefore executes the first action in the plan that minimizes the *expected* cost: $\pi^j(x^j_t) = \pi^{*,j}_{t|t}(x^j_{t|t})$.

Definition IV.3. *The expected cost of π^j at iteration j from start state x_0 is defined as:*

$$J^{\pi^j}(x^j_0) = \mathbb{E}_{x^j} \left[\sum_{t=0}^{\infty} C(x^j_t, \pi_j(x^j_t)) \right] = V^{\pi^j}(x^j_0) \tag{IV-A.7}$$

Definition IV.4. *Robust Control Invariant Set: As in Rosolia* et al. *[27], we define a robust control invariant set $\mathcal{A} \subseteq \mathcal{X}$ with respect to dynamics $f(x,u,w)$ and policy class Π as a set where $\forall x \in \mathcal{A}$, $\exists \pi \in \Pi$ s.t. $f(x, \pi(x), w) \in \mathcal{A}$, $\forall w \in \mathcal{W}$.*

Assumption IV.3. *Robust Control Invariant Goal Set: $\mathcal{G}$ is a robust control invariant set with respect to the dynamics and policy class.*

Assumption IV.4. *Robust Control Invariant Sampled Safe Set: We assume that $\mathcal{SS}^j$ is a robust control invariant set with respect to the dynamics and policy class for all j. Since $x_0 \in \mathcal{SS}^j \ \forall j$, note that this implies that $J^j_{0 \to H}(x^j_0) < \infty \ \forall j$.*

It can be shown that Assumptions IV.3 and IV.4 hold in the limit of infinite samples from the control policy at each iteration [27]. This is intuitive, since in the limit of infinite samples, we sample every possible noise realization. The amount of data needed to approximately meet these assumptions in practice is related to environmental stochasticity.

Assumption IV.5. *Constant Start State. The start state x_0 is constant across iterations.*

Assumption IV.5 is reasonable in the settings we consider, since the start state distribution has low variance in all experiments. The analysis is easily extended to non-constant start states, but practically requires more data to satisfy assumption IV.4, especially for wider start state distributions.

Assumption IV.6. *Completion Cost Specification. We assume that $\exists \varepsilon > 0$ s.t. $C(x,\cdot) \geq \varepsilon \mathbb{1}_{\mathcal{G}^C}(x)$ and $C(x,\cdot) = 0 \ \forall x \in \mathcal{G}$*

Note that assumption IV.6 holds for all experiments, since costs are specified as above with equality and $\varepsilon = 1$.

B. SAVED Convergence Analysis

The main contribution of the following analysis is to show that the proposed control strategy guarantees iterative improvement of expected performance for known stochastic nonlinear systems. We emphasize that Assumptions IV.1-IV.5 are standard as in [27] and the only extra assumption is assumption IV.6. See supplementary material for all proofs.

Lemma IV.1. *Recursive Feasibility: Consider system* (IV-A.1) *in closed-loop with* (IV-A.4). *Let the sampled safe set $\mathcal{SS}^j$ be defined as in* (III-C.2). *If assumptions IV.1-IV.6 hold, then the controller* (IV-A.3) *and* (IV-A.4) *is feasible for $t \geq 0$ and $j \geq 0$ in expectation. Equivalently, $\mathbb{E}_{x^j_t}[J^j_{t \to t+H}(x^j_t)] < \infty$.*

Lemma IV.1 shows that SAVED is expected to satisfy state-space constraints for all timesteps t in all iterations j.

Lemma IV.2. *Convergence in Probability: Consider the closed-loop system* (IV-A.1) *and* (IV-A.4). *Let $\mathcal{SS}^j$ be defined as in* (III-C.2) *and assumptions IV.1-IV.6 hold. If the closed-loop system converges in probability to $\mathcal{G}$ at the initial iteration, then it converges in probability at all subsequent iterations. Precisely, at iteration j: $\lim_{t \to \infty} P(x^j_t \notin \mathcal{G}) = 0$.*

Theorem IV.1. *Iterative Improvement: Consider system* (IV-A.1) *in closed-loop with* (IV-A.4). *Let the sampled safe set $\mathcal{SS}^j$ be defined as in* (III-C.2). *Given assumptions IV.1-IV.6, the expected cost-to-go (IV-A.7) associated with control policy* (IV-A.4) *is non-increasing in iterations:*

$$\forall j \in \mathbb{N}, \ J^{\pi^j}(x_0) \geq J^{\pi^{j+1}}(x_0)$$

Furthermore, $\{J^{\pi^j}(x_0)\}_{j=0}^{\infty}$ is a convergent sequence.

Theorem IV.1 is an interesting new theoretical result, because while past work has provided similar guarantees for robust controllers in stochastic linear systems or deterministic nonlinear systems [26–28] under similar assumptions, we provide iterative improvement guarantees in expectation for stochastic nonlinear systems with task completion costs.

V. Experiments

We evaluate SAVED on simulated continuous control benchmarks and on real robotic tasks with the da Vinci Research Kit (dVRK) [14] against state-of-the-art deep RL algorithms and demonstrate that SAVED outperforms all baselines in terms of sample efficiency, success rate, and constraint satisfaction during learning. All tasks use $C(x,u) = \mathbb{1}_{\mathcal{G}^C}(x)$ (Section III-A), which is equivalent to the time spent outside the goal set. All algorithms are given the same demonstrations, are evaluated on iteration cost, success rate, and constraint satisfaction rate (if applicable), and run 3 times to control for stochasticity in training. Tasks are only considered successfully completed if the agent reaches and stays in $\mathcal{G}$ until the end of the episode without ever violating constraints. For all simulated tasks, we give model-free methods 10,000 iterations since they take much longer to converge but sometimes have better asymptotic performance. See supplementary material for additional experiments, videos, and ablations with respect to choice of α, β, and demonstration quantity. We also include further details on baselines, network architectures, hyperparameters, and training procedures.

A. Baselines

We consider the following set of model-free and model-based baseline algorithms. To enforce constraints for model-based baselines, we augment the algorithms with the simulation based method described in Section III-D. Because model-free baselines have no such mechanism to readily enforce constraints, we instead apply a very large cost when constraints are violated. See supplementary material for an ablation of the reward function used for model-free baselines.

1) **Behavior Cloning (Clone)**: Supervised learning on demonstrator trajectories.
2) **PETS from Demonstrations (PETSfD)**: Probabilistic ensemble trajectory sampling (PETS) from Chua et al [7] with the dynamics model initialized with demo trajectories and planning horizon long enough to plan to the goal (judged by best performance of SAVED).
3) **PETSfD Dense**: PETSfD with tuned dense cost.
4) **Soft Actor Critic from Demonstrations (SACfD)**: Model-free RL algorithm, Soft Actor Critic [12], where demo transitions are used for training initially.
5) **Overcoming Exploration in Reinforcement Learning from Demonstrations (OEFD)**: Model-free algorithm from Nair *et al.* [22] which combines model-free RL with a behavior cloning loss to accelerate learning.
6) **SAVED (No SS)**: SAVED without the *sampled safe set* constraint described in Section III-C.

B. Simulated Navigation

To demonstrate if SAVED can efficiently and safely learn temporally extended tasks with complex constraints, we consider a set of tasks in which a point mass navigates to a unit ball centered at the origin. The agent can exert force in cardinal directions and experiences drag and Gaussian process noise in the dynamics. For each task, we supply 50 to 100 suboptimal demonstrations, generated by running LQR along a hand-tuned safe trajectory. SAVED has a higher success rate than all other RL baselines using sparse costs, even including model-free baselines over the first 10,000 iterations, while never violating constraints across all navigation tasks. Furthermore, this performance advantage is amplified with task difficulty. Only Clone and PETSfD Dense ever achieve a higher success rate, but Clone does not improve upon demonstration performance (Figure 3) and PETSfD Dense has additional information about the task. Furthermore, SAVED learns significantly more efficiently than all RL baselines on all navigation tasks except for tasks 1 and 3, in which PETSfD Dense with a Euclidean norm cost function finds a better solution. While SAVED (No SS) can complete the tasks, it has a much lower success rate than SAVED, especially in environments with obstacles as expected, demonstrating the importance of the *sampled safe set* constraint. Note that SACfD, OEFD, and PETSfD make essentially no progress in the first 100 iterations and never complete any of the tasks in this time, although they mostly satisfy constraints.

C. Simulated Robot Experiments

To evaluate whether SAVED also outperforms baselines on standard unconstrained environments, we consider sparse versions of two common simulated robot tasks: the PR2 Reacher environment used in Chua *et al.* [7] with a fixed goal and on a pick and place task with a simulated, position-controlled Fetch robot. The reacher task involves controlling the end-effector of a simulated PR2 robot to a small ball in $\mathbb{R}^3$. The pick and place task involves picking up a block from a fixed location on a table and also guiding it to a small ball in $\mathbb{R}^3$. The task is simplified by automating the gripper motion, which is difficult for SAVED to learn due to the bimodality of gripper controls, which is hard to capture with the unimodal truncated Gaussian distribution used during CEM sampling. SAVED still learns faster than all baselines on both tasks (Figure 4) and exhibits significantly more stable learning in the first 100 and 250 iterations for the reacher and pick and place tasks respectively.

D. Physical Robot Experiments

We evaluate the ability of SAVED to learn a surgical knot-tying task with nonconvex state space constraints on the da Vinci Research Kit (dVRK) [14]. The dVRK is cable-driven and has relatively imprecise controls, motivating model learning [29]. Furthermore, safety is paramount due to the cost and delicate structure of the arms. The goal here is to speed up demo trajectories by constraining learned trajectories to fall within a tight, 1 cm tube of the demos, making this difficult for many RL algorithms. Additionally, robot experiments are very time consuming, so training RL algorithms on limited physical hardware is difficult without sample efficient algorithms. We also include additional experiments on a Figure-8 tracking task in the supplementary material.

1) Surgical Knot-Tying: SAVED is used to optimize demonstrations of a surgical knot-tying task on the dVRK, using the same multilateral motion as in [32]. Demonstrations

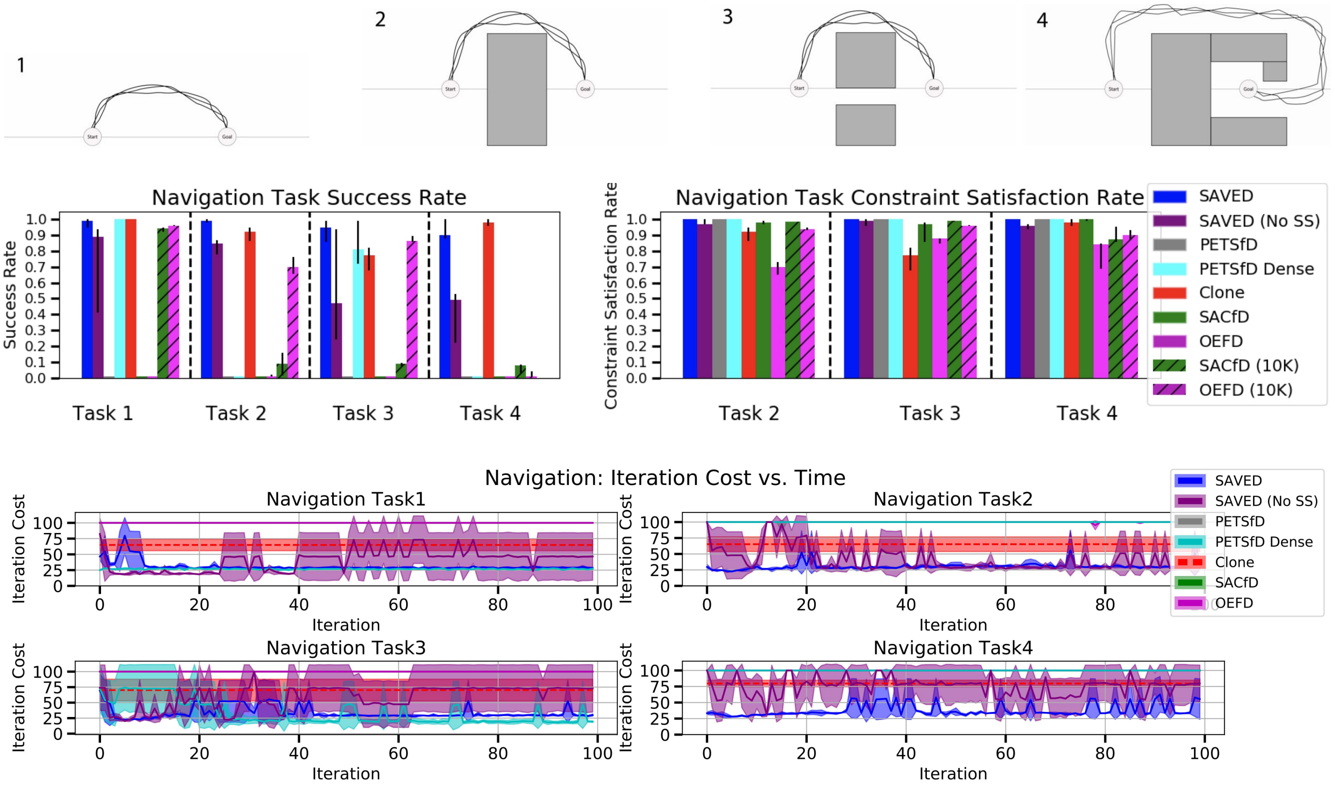

Fig. 3: **Navigation Domains:** SAVED is evaluated on 4 navigation tasks. Tasks 2-4 contain obstacles, and task 3 contains a channel for passage to $\mathcal{G}$ near the x-axis. SAVED learns significantly faster than all RL baselines on tasks 2 and 4. In tasks 1 and 3, SAVED has lower iteration cost than baselines using sparse costs, but does worse than PETSfD Dense, which is given dense Euclidean norm costs to find the shortest path to the goal. For each task and algorithm, we report success and constraint satisfaction rates over the first 100 training iterations and also over the first 10,000 iterations for SACfD and OEFD. We observe that SAVED has higher success and constraint satisfaction rates than other RL algorithms using sparse costs across all tasks, and even achieves higher rates in the first 100 training iterations than model-free algorithms over the first 10,000 iterations.

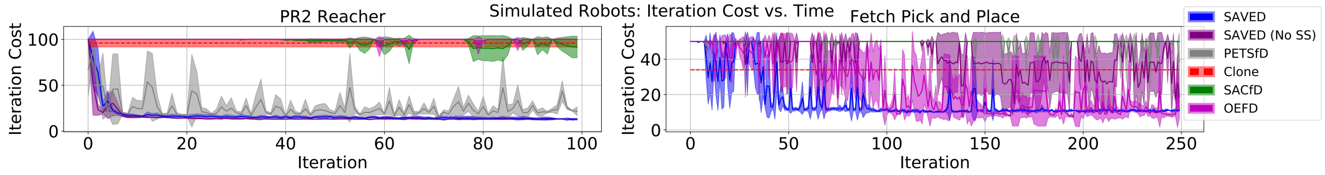

Fig. 4: **Simulated Robot Experiments Performance:** SAVED achieves better performance than all baselines on both tasks. We use 20 demonstrations with average iteration cost of 94.6 for the reacher task and 100 demonstrations with average iteration cost of 34.4 for the pick and place task. For the reacher task, the safe set constraint does not improve performance, likely because the task is very simple, but for pick and place, we see that the safe set constraint adds significant training stability.

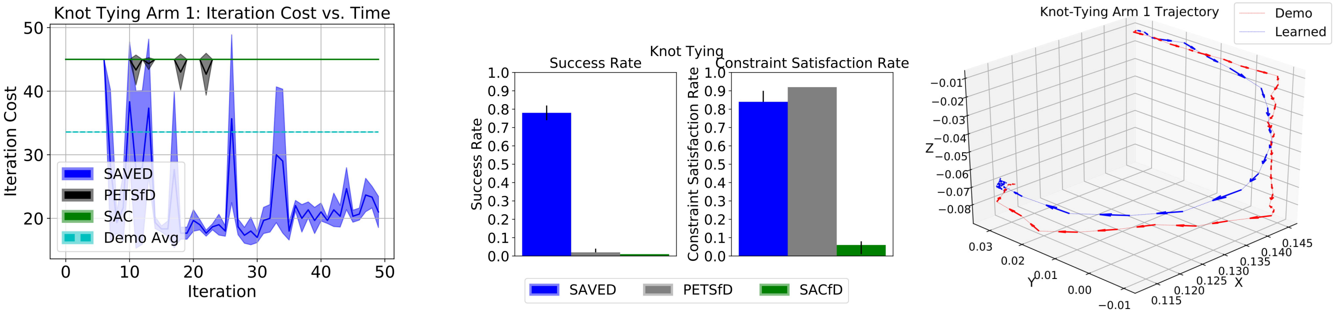

Fig. 5: **Surgical Knot-Tying: Training Performance:** After just 15 iterations, the agent completes the task relatively consistently with only a few failures, and converges to a iteration cost of 22, faster than demos, which have an average iteration cost of 34. In the first 50 iterations, both baselines mostly fail, and are less efficient than demos when they do succeed; **Trajectories:** SAVED quickly learns to speed up with only occasional constraint violations.

are hand-engineered for the task, and then policies are optimized for one arm (arm 1), while a hand-engineered policy is used for the other arm (arm 2). We do this because while arm 1 wraps the thread around arm 2, arm 2 simply moves down, grasps the other end of the thread, and pulls it out of the phantom as shown in Figure 1. Thus, we only expect significant performance gain by optimizing the policy for the portion of the arm 1 trajectory which involves wrapping the thread around arm 2. We only model the motion of the end-effectors in 3D space. SAVED quickly learns to smooth out demo trajectories, with a success rate of over 75% (Figure 5) during training, while baselines are unable to make sufficient progress in this time. PETSfD rarely violates constraints, but also almost never succeeds, while SACfD almost always violates constraints and never completes the task. Training SAVED directly on the real robot for 50 iterations takes only about an hour, making it practical to train on a real robot for tasks where data collection is expensive. At execution-time, we find that SAVED is very consistent, successfully tying a knot in $20/20$ trials with average iteration cost of 21.9 and maximum iteration cost of 25 for the arm 1 learned policy, significantly more efficient than demos which have an average iteration cost of 34. See supplementary material for trajectory plots of the full knot-tying trajectory and the figure 8 task.

VI. Discussion and Future Work

We present SAVED, a model-based RL algorithm that can efficiently learn a variety of robotic control tasks in the presence of dynamical uncertainty, sparse cost feedback, and complex constraints. SAVED uses a small set of sub-optimal demonstrations and a learned state-value function and constrains exploration to regions in which the agent is confident. We present iterative improvement guarantees in expectation for SAVED for stochastic nonlinear systems. We empirically evaluate SAVED on 6 simulated benchmarks and on a knot-tying task on a real surgical robot. Results suggest that SAVED is more sample efficient and has higher success and constraint satisfaction rates than all RL baselines and can be efficiently and safely trained on a real robot. We believe this work opens up opportunities to further study safe RL, specifically for visual and multi-goal planning.

References

[1] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization", in *Journal of Machine Learning Research*, 2017.

[2] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in AI safety", *arXiv preprint arXiv:1606.06565*, 2016.

[3] F. Berkenkamp, M. Turchetta, A. P. Schoellig, and A. Krause, "Safe model-based reinforcement learning with stability guarantees", in *NIPS*, 2017.

[4] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[5] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control", *IEEE control systems magazine*, 2006.

[6] D. S. Brown, W. Goo, P. Nagarajan, and S. Niekum, "Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations", vol. abs/1904.06387, 2019.

[7] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models", in *Proc. Advances in Neural Information Processing Systems*, 2018.

[8] M. Deisenroth and C. Rasmussen, "PILCO: A model-based and data-efficient approach to policy search", in *Proc. Int. Conf. on Machine Learning*, 2011.

[9] J. Fu, S. Levine, and P. Abbeel, "One-shot learning of manipulation skills with online dynamics adaptation and neural network priors", in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2016.

[10] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods", in *Proc. Int. Conf. on Machine Learning*, 2018.

[11] J. García and F. Fernández, "A comprehensive survey on safe reinforcement learning", *Journal of Machine Learning Research*, 2015.

[12] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor", in *Proc. Int. Conf. on Machine Learning*.

[13] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband, *et al.*, "Deep q-learning from demonstrations", in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[14] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da Vinci surgical system", in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2014.

[15] I. Lenz, R. A. Knepper, and A. Saxena, "DeepMPC: Learning deep latent features for model predictive control", in *Robotics: Science and Systems*, 2015.

[16] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies", *Journal of Machine Learning Research*, 2016.

[17] Z. Li, U. Kalabić, and T. Chu, "Safe reinforcement learning: Learning with supervision using a constraint-admissible set", in *2018 Annual American Control Conference (ACC)*, 2018.

[18] K. Lowrey, A. Rajeswaran, S. Kakade, E. Todorov, and I. Mordatch, "Plan online, learn offline: Efficient learning and exploration via model-based control", in *Proc. Int. Conf. on Machine Learning*, 2019.

[19] T. M. Moldovan and P. Abbeel, "Risk Aversion in Markov Decision Processes via near optimal Chernoff bounds", in *Proc. Advances in Neural Information Processing Systems*, 2012.

[20] T. M. Moldovan and P. Abbeel, "Safe exploration in markov decision processes", *arXiv preprint arXiv:1205.4810*, 2012.

[21] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning", in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2018.

[22] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations", *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2018.

[23] A. Nemirovski, "On safe tractable approximations of chance constraints", *European Journal of Operational Research*, 2012.

[24] T. Osogami, "Robustness and risk-sensitivity in markov decision processes", in *NIPS*, 2012.

[25] U. Rosolia, A. Carvalho, and F. Borrelli, "Autonomous racing using learning model predictive control", in *Proceedings 2017 IFAC World Congress*, 2017.

[26] U. Rosolia and F. Borrelli, "Learning model predictive control for iterative tasks. a data-driven control framework", *IEEE Transactions on Automatic Control*, 2018.

[27] ——, "Sample-based learning model predictive control for linear uncertain systems", *CoRR*, vol. abs/1904.06432, 2019. arXiv: `1904.06432`.

[28] U. Rosolia, X. Zhang, and F. Borrelli, "A Stochastic MPC Approach with Application to Iterative Learning", *2018 IEEE Conference on Decision and Control (CDC)*, 2018.

[29] D. Seita, S. Krishnan, R. Fox, S. McKinley, J. Canny, and K. Goldberg, "Fast and reliable autonomous surgical debridement with cable-driven robots using a two-phase calibration procedure", in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2018.

[30] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st. Cambridge, MA, USA: MIT Press, 1998.

[31] S. Tu and B. Recht, "The gap between model-based and model-free methods on the linear quadratic regulator: An asymptotic viewpoint", *CoRR*, vol. abs/1812.03565, 2018.

[32] J. Van Den Berg, S. Miller, D. Duckworth, H. Hu, A. Wan, X.-Y. Fu, K. Goldberg, and P. Abbeel, "Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations", in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2010.

[33] M. Vecerik, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. Heess, T. Rothörl, T. Lampe, and M. A. Riedmiller, "Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards", *CoRR*, vol. abs/1707.08817, 2017.