This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2020.3015931, IEEE Transactions on Automatic Control

Constrained Cross-Entropy Method for Safe Reinforcement Learning

Min Wen and Ufuk Topcu

***Abstract*—We study a safe reinforcement learning problem in which the constraints are defined as the expected cost over finite-length trajectories. We propose a constrained cross-entropy-based method to solve this problem. The key idea is to transform the original constrained optimization problem into an unconstrained one with a surrogate objective. The method explicitly tracks its performance with respect to constraint satisfaction and thus is well-suited for safety-critical applications. We show that the asymptotic behavior of the proposed algorithm can be almost-surely described by that of an ordinary differential equation. Then we give sufficient conditions on the properties of this differential equation for the convergence of the proposed algorithm. At last we show the performance of the proposed algorithm in two simulation examples. In a constrained linear quadratic regulator example, we observe that the algorithm converges to the global optimum with high probability. In a 2D navigation example, we find the algorithm effectively learn feasible policies without assumptions on the feasibility of initial policies, even with non-Markovian objective functions and constraint functions.**

I. Introduction

We study the following constrained optimal control problem in this paper: Given a dynamical system model with continuous states and actions, a objective function and a constraint function, find a controller that maximizes the objective function while satisfying the constraint. Although this topic has been studied for decades within the control community [1], it is still challenging for practical problems. To illustrate some major difficulties, consider the synthesis of a policy for a nonholonomic mobile robot to reach a goal while avoiding obstacles (which introduces constraints) in a cost-efficient way (which induces an objective). The obstacle-free state space is usually nonconvex. The equations of the dynamical system model are typically highly nonlinear. Constraint functions and cost functions may not be convex or differentiable in the state and action variables. There may even be hidden variables that are not observable and make transitions and costs non-Markovian. Given all these difficulties, we still need to compute a policy that is at least feasible and improve the cost objective as much as possible.

Reinforcement learning (RL) methods have been widely used to learn optimal policies for agents with complicated or even unknown dynamics. For problems with continuous state and action spaces, the agent's policy is usually modeled as a parameterized function of states such as deep neural networks and later trained using policy gradient methods [2], [3], [4], [5], [6], [7], [8]. By encoding control tasks as reward or cost functions, RL has successfully solved a wide range of tasks such as Atari games [9], [10], the game of Go [11], [12], controlling simulated robots [13], [14] and real robots [15], [16], [17].

Most of the existing methods for RL solve only unconstrained problems. However, it is generally non-trivial to transform a constrained optimal control problem into an unconstrained one, due to the asymmetry between the goals of objective optimization and constraint satisfaction. On the one hand, it is usually acceptable to output a policy that is only locally optimal with respect to the optimization objective. On the other hand, in many application scenarios where constraints encode safety requirements or the amount of available resources, violating the constraint even by a small amount may have significant consequences.

Existing methods for safe RL that are based on policy gradient methods cannot guarantee strict feasibility of the policies they output, even when initialized with feasible initial policies. When initialized with an infeasible policy, they usually are not be able to find even a single feasible policy until their convergence (with an example in Section V). These limitations motivate the following question: Can we develop an RL algorithm that explicitly addresses the priority of constraint satisfaction? Rather than assuming that the initial policy is feasible and that one can always find a feasible policy in the estimated gradient direction, we need to deal with cases in which the initial policy is not feasible, or we have never seen a feasible policy before.

Inspired by stochastic optimization methods based on the cross-entropy (CE) concept [18], we propose a new safe RL algorithm, which we call the *constrained cross-entropy (CCE)* method. The basic framework is the same with standard CE methods: In each iteration, we sample from a distribution of policies, select a set of elite sample policies and use them to update the policy distribution. Rather than treating the constraints as an extra term in the objective function as what policy gradient method do, we use constraint values to sort sample policies. If there are not enough feasible sample policies, we select only those with the best constraint performance as elite sample policies. If a given proportion of the sample policies are feasible, we select the feasible sample policies with the best objective values as elite sample policies. Instead of initializing the optimization with a feasible policy, the method improves both the objective function and the constraint function with the constraint as a prioritized concern.

Our algorithm can be used as a black-box optimizer. It does

The work was done while M. Wen was with the Department of Electrical and Systems Engineering, University of Pennsylvania. She is now working at Google LLC. Email: minwen@google.com.

U. Topcu is with the Department of Aerospace Engineering and Engineering Mechanics, University of Texas at Austin, Austin, TX, 78712, USA. Email: utopcu@utexas.edu.

not even assume that there is an underlying reward or cost function encoding the optimization objective and constraint functions. In fact, the algorithm can be applied to any finite-horizon problem (say, with horizon N) whose objective and constraint functions are defined as the average performance over some distribution of trajectories. For example, a constraint function can be the probability that the agent satisfies a given task specification (which may be Markovian or non-Markovian) with policy π_θ, if the satisfaction of the given task can be decided with any N-step trajectory. An optimization objective may be the expected number of steps before the agent reaches a goal state, or the expected maximum distance the agent has left from its origin, or the expected minimum distance between the agent and any obstacle over the whole trajectory.

Our contributions are as follows. First, we present a model-free constrained RL algorithm that works with continuous state and action spaces and is very simple to implement. Second, we prove that the asymptotic behavior of our algorithm can be almost-surely described by that of an ordinary differential equation (ODE), which is easily interpretable with respect to the objectives. Third, we give sufficient conditions on the properties of this ODE to guarantee the convergence of our algorithm. At last, we empirically show that our algorithm converges to the global optimum with high probability in a convex problem, and effectively find feasible policies in a 2D navigation example while other policy-gradient-based algorithms fail to find strictly feasible solutions.

II. Related Work

It has been a long-lasting interest to incorporate safety into RL process [19], while the definition of safety depends on each specific work. For example, safety may refer to avoiding unsafe states during exploration [20], [21], [22], [23], [24], the stability of the closed-loop system [25], or monotonic improvement in policy values [26]. We choose to use the so-called *constrained criterion* [19] to encode our safety requirement, such that safety is ensured if and only if the expectation of some constraint measure is upper bounded by a certain threshold. The goal is to find a feasible policy that solves a constrained optimization problem [27], [28], [29], although some works can also provide consistent feasibility guarantees during exploration with some extra assumptions [30], [31].

With the constrained criterion, it is common to model safe RL problems as constrained Markov decision processes (CMDP) [32], where the constraint measure is defined with a transition-based constraint cost. There are many ways to incorporate the constraint condition into policy updates. One way is to use the Lagrangian method, which is a standard approach to solve constrained optimization problems. For example, Chow et al. [28] came up with a trajectory-based primal-dual subgradient algorithm for a risk-constrained RL problem with finite state and action spaces. The algorithm is proved to converge almost-surely to a local saddle point. However, the algorithm requires solving a series of unconstrained RL problems, which is impractical for problems with large state spaces or continuous states.

Another way to enforce constraint satisfaction is by projection, where policies are first updated without considering constraints and then projected to the feasible region. For example, Uchibe and Doya [27] proposed a constrained policy gradient algorithm which relies on projected gradients to maintain feasibility. The computation of projection restricts the types of constraints it can deal with, and there is no known guarantee on convergence. Despite policy parameters, projection can also be used to directly map actions. Dalal et al. [33] introduced a *safety layer* that perturbs the original action from policy at each state if necessary, which can preserve feasibility if the constraints are evaluated on transitions. Chow et al. [31] further applied the idea of safety layers to problems with trajectory-based constraints, where trajectory-based constraints are broken into sequences of single-step state dependent constraints via a Lyapunov-based approach [30].

There are also works that handle constraints by estimating value functions and bounding the difference caused by each policy update. For example, Achiam et al. [34] limits the policy updates within trust regions where the change in constraint values over a single policy update is bounded by a function of the advantage function of the previous policy. Yu et al. [29] transform the original (non-convex) constrained optimization problem into a sequence of surrogate convex constrained problems.

Cross-entropy-based stochastic optimization techniques have been applied to a series of RL and optimal control problems. Mannor, Rubinstein and Gat [35] used CE methods to solve a stochastic shortest-path problem on finite Markov decision processes, which is essentially an unconstrained problem. Szita and Lörincz [36] took a noisy variant to learn how to play Tetris. Kobilarov [37] introduced a similar technique to motion planning in constrained continuous-state environments by considering distributions over collision-free trajectories. Livingston, Wolff and Murray [38] generalized this method to deal with a broader class of trajectory-based constraints called linear temporal logic specifications. Both methods simply discard all sample trajectories that violate the given constraints, and thus their work can be considered as a special case of our work when the constraint function has binary outputs. Similar applications in approximate optimal control with constraints can be found in [39], [40], [41].

III. Preliminaries

We first introduce some notations that are used throughout this paper. For a set B, let $\mathcal{D}(B)$ be the set of all probability distributions over B, $int(B)$ be the interior of B and $\mathbf{1}_B$ be the indicator function of B. For any $k \in \mathbb{N}^+$, define $B^k := \{s_0, s_1, \ldots, s_{k-1} \mid s_t \in B,\ \forall t = 0, \ldots, k-1\}$ as the set of all sequences composed by elements in B of length k. We further define $B^* := \bigcup_{1 \le k < \infty} B^k$ as the set of all (non-empty) finite sequences generated by elements in B. Given two integers $i, j \in \mathbb{N}$ such that $i \le j$, we use $i : j$ to denote the sequence $i, i+1, \ldots, j-1, j$.

A (reward-free) *Markov decision process (MDP)* is defined as a tuple $M = \langle S, A, T, P_0 \rangle$, where S is a set of states, A is a set of actions, $T : S \times A \to \mathcal{D}(S)$ is a transition distribution function and $P_0 \in \mathcal{D}(S)$ is an initial state distribution. Without loss of generality, we assume that the set of available actions are the same at all states. S and A can either be continuous or discrete.

Given an MDP M, a *policy* $\pi : S^* \to \mathcal{D}(A)$ is a mapping from a sequence of history states to a distribution over actions. π is called *stationary* or *memoryless* if its output is decided by the last state in history, that is, $\pi(\zeta) = \pi(s_k)$ holds for any $\zeta = \zeta_0, \zeta_1, \ldots, \zeta_k \in S^*$. π is called *deterministic* if the support of its output distribution is always a singleton. For notational simplicity, we use $\pi(\zeta)$ to represent the unique action $a \in A$ such that $\pi(a|\zeta) > 0$ for any $\zeta \in S^*$. If π is not deterministic, we call it a *randomized* policy. Let Π, Π_S, Π_D, Π_{SD} be the set of all policies, stationary policies, deterministic policies and stationary deterministic policies for M. It is clear that $\Pi_{SD} = \Pi_S \bigcap \Pi_D \subset \Pi$.

Given a finite horizon $N \in \mathbb{N}^+$, an *N-step trajectory* τ is a sequence of N state-action pairs: $\tau = s_0, a_0, \ldots, s_{N-1}, a_{N-1} \in (S \times A)^N$. Each policy $\pi \in \Pi$ decides a distribution $P_{\pi,N}$ over N-step trajectories such that for any $\tau = s_0, a_0, \ldots, s_{N-1}, a_{N-1}$, $P_{\pi,N}(\tau) = P_0(s_0) \prod_{t=0}^{N-2} T(s_{t+1}|s_t, a_t) \prod_{t=0}^{N-1} \pi(a_t|s_{0:t})$. Without loss of generality, we assume that N is fixed and use P_π to represent $P_{\pi,N}$.

To solve an N-step planning problem, we can generally define a trajectory-based *objective* function $J : (S \times A)^N \to \mathbb{R}$ as a mapping from each N-step trajectory to a scalar value. For each $\pi \in \Pi$, let

$$G_J(\pi) := \mathbb{E}_{\tau \sim P_\pi}[J(\tau)]$$

be the expected value of J with the N-step trajectory distribution decided by π. Many commonly used objectives for finite-horizon planning problems can be represented as G_J, such as

- Expected N-step total reward. Given a reward function $R : S \times A \to \mathbb{R}$, define $J(\tau) = \sum_{t=0}^{N-1} R(s_t, a_t)$ for $\tau = s_0, a_0, \ldots, s_{N-1}, a_{N-1}$ and $G_J(\pi)$ will be the expected N-step total reward while running π.
- Probability. Given a set of N-step trajectories $B \subseteq (S \times A)^N$, define $J(\tau) = \mathbf{1}_B(\tau)$ and $G_J(\pi)$ will be the probability to induce a trajectory in B while running π. For example, $G_J(\pi)$ can be used to represent the probability to reach a set of target states or the probability to remain in a safe region for N steps.

A policy $\pi^* \in \Pi$ is *optimal* with respect to J if $G_J(\pi^*) = \max_{\pi' \in \Pi} G_J(\pi')$. Generally, π^* is not stationary if the horizon N is finite. But since the transition distribution $\mathcal{T}$ is Markovian, there always exists a (non-stationary) deterministic optimal policy, which is formally stated in Lemma 1.

Though the proof below is given for models with continuous state and action spaces, it can be easily adapted to discrete models.

Lemma 1. *Given $N \in \mathbb{N}^+$ be a finite horizon and an MDP M, let $J : (S \times A)^N \to \mathbb{R}$ be any bounded trajectory-based functional. There always exists a deterministic (yet possibly non-stationary) optimal policy π^*. In other words, there exists $\pi_d \in \Pi_D$ such that*

$$G_J(\pi_d) = \max_{\pi \in \Pi} G_J(\pi).$$

Proof. Let $\pi : S^* \to \mathcal{D}(A)$ be a policy for M. Then it generates a distribution over N-step trajectories which is P_π. For any $t = 0, \ldots, N-1$, the probability that $s_0, a_0, \ldots, s_t$ (denoted as $s_{0:t}, a_{0:t-1}$) is a prefix of a generated trajectory is

$$\begin{aligned}
&p_t^{pre} := P_\pi(s_{0:t}, a_{0:t-1}) \\
&= \int P_\pi(s_{0:t}, a_{0:t-1}, s'_{t+1:N-1}, a'_{t:N-1}) da'_{t:N-1}, s'_{t+1:N-1} \\
&= P_0(s_t) \prod_{t'=0}^{t-1} \Big(\pi(a_{t'} \mid s_{0:t'}) T(s_{t'+1} \mid s_{t'}, a_{t'})\Big).
\end{aligned}$$

Given a prefix $s_{0:t}, a_{0:t}$, the probability that the next $(N-t-1)$ state-action pairs are $s_{t+1}, a_{t+1}, \ldots, s_{N-1}, a_{N-1}$ (denoted as $s_{t+1:N-1}, a_{t+1:N-1}$) is

$$\begin{aligned}
p_{t+1}^{suf} :=& P_\pi\big(s_{t+1:N-1}, a_{t+1:N-1} \mid s_{0:t}, a_{0:t}\big) \\
=& \prod_{t'=t}^{N-2} T(s_{t'+1} \mid s_{t'}, a_{t'}) \prod_{t'=t+1}^{N-1} \pi(a_{t'} \mid s_{0:t'}).
\end{aligned}$$

Define $J_N := J(s_{0:N-1}, a_{0:N-1})$. We can rewrite $G_J(\pi)$ as

$$G_J(\pi) = \int \pi(a_t \mid s_{0:t}) p_t^{pre} p_{t+1}^{suf} J_N \ ds_{0:N-1}, a_{0:N-1}.$$

Since $\int \pi(a_t \mid s_{0:t}) p_t^{pre} p^{suf} \ ds_{0:N-1}, a_{0:N-1} = 1$ and J is bounded, $\int |\pi(a_t \mid s_{0:t}) p_t^{pre} p^{suf} J_N| \ ds_{0:N-1}, a_{0:N-1} < \infty$. Therefore by Fubini's Theorem,

$$\begin{aligned}
&G_J(\pi) \\
&= \int \pi(a_t \mid s_{0:t}) p_t^{pre} p_{t+1}^{suf} J_N \cdot \\
&\quad ds_{0:t}, s_{t+1:N-1}, a_{0:t-1}, a_t, a_{t+1:N-1} \\
&= \int \pi(a_t \mid s_{0:t}) \big(\int p_t^{pre} p_{t+1}^{suf} J_N ds_{t+1:N-1}, a_{0:t-1}, a_{t+1:N-1} \big) \cdot \\
&\quad ds_{0:t}, a_t \\
&= \int \pi(a_t \mid s_{0:t}) Q_\pi(a_t \mid s_{0:t}) ds_{0:t}, a_t,
\end{aligned}$$

where we define

$$Q_\pi(a_t \mid s_{0:t}) := \int p_t^{pre} p_{t+1}^{suf} J_N ds_{t+1:N-1}, a_{0:t-1}, a_{t+1:N-1}.$$

Note that p_t^{pre}, p_{t+1}^{suf} and J_N are independent on $\pi(a_t|s_{0:t})$; $\pi(a_t \mid s_{0:t})$ is also independent for different t and prefix $s_{0:t}$. Therefore $Q_\pi(a \mid s_{0:t})$ is independent on $\pi(a \mid s_{0:t})$. For any prefix except $s_{0:t}$, it holds for any optimal policy $\pi'(\cdot \mid s_{0:t})$ that maximizes G_J that $\{a \in A \mid \pi'(a \mid s_{0:t}) > 0\} \subseteq \arg\max_{a \in A} Q_\pi(a \mid s_{0:t})$, which always incorporates a deterministic choice. In other words, randomized policies cannot reach higher G_J than deterministic policies. □

Similarly, we can define a trajectory-based *cost* function $Z : (S \times A)^N \to \mathbb{R}$ and define

$$H_Z(\pi) := \mathbb{E}_{\tau \sim P_\pi}[Z(\tau)]$$

as the expected cost over trajectory distribution P_π. A policy $\pi \in \Pi$ is *feasible* for a constrained optimization problem with cost function Z and *constraint upper bound* d if $H_Z(\pi) \leq d$.

Note that since Z can be *any* trajectory-based cost function and may not be represented as the sum of a transition-based cost function, our problem is more general than finite-horizon CMDP problems.

For notational simplicity, we omit J and Z in G_J and H_Z whenever there is no ambiguity. For any policy $\pi \in \Pi$, we refer to $G(\pi)$ and $H(\pi)$ as the *G-value* and *H-value* of π.

IV. Constrained Cross-Entropy Framework

In this section, we first state the constrained policy optimization given a trajectory-based objective function J and a trajectory-based cost function Z, then we describe how to transform the constrained problem into an unconstrained one with a surrogate objective function. We propose an algorithm called constrained cross-entropy method to optimize the surrogate objective and show that the algorithm converges almost surely with some given assumptions.

A. Problem Formulation

In this paper, we consider a finite-horizon RL problem with a strictly positive objective function $J : (S \times A)^N \to \mathbb{R}^+$, a cost function $Z : (S \times A)^N \to \mathbb{R}$ and a constraint upper bound d. For MDPs with continuous state and action spaces, it is usually intractable to exactly solve an optimal stationary policy due to the curse of dimensionality. An alternative is to use function approximators, such as neural networks, to parameterize a subset of policies. Given a parameterized class of policies Π_Θ with a parameter space $\Theta \subseteq \mathbb{R}^{d_\theta}$, we aim to solve the following problem:

$$\pi^* = \operatorname*{arg\,max}_{\pi \in \Pi_\Theta \bigcap \Pi_{Z,d}} [G_J(\pi)], \tag{1}$$

where $\Pi_{Z,d} = \{\pi \in \Pi | H_Z(\pi) \leq d\}$ is the set of feasible policies.

The proposed algorithm, which we call the *constrained cross-entropy* method, generalizes the well-known cross-entropy method [35] for unconstrained optimization. The basic idea is to generate a sequence of policy distributions that eventually concentrates on a feasible (locally) optimal policy. Given a distribution over Π_Θ, we randomly generate a set of sample policies, sort them with a ranking function that depends on their G-values and H-values and then update the policy distribution with a subset of highly ranked sample policies. The set of sample policies that are selected to update the current policy distribution are also referred to as *elite samples* or *elite set* in the literature (for example, [35], [37], [38]).

Given the policy parameterization Π_Θ, we use distributions over the parameter space Θ to represent distributions over the policy space Π_Θ. Let $f : \mathcal{V} \to \mathcal{D}(\Theta)$ be a family of distributions over Θ with parameter space $\mathcal{V}$. For each $\boldsymbol{v} \in \mathcal{V}$, $f_{\boldsymbol{v}}(\cdot)$ is a distribution over policies in Π_Θ. We assume that for any $\theta \in \Theta$, there exists $\boldsymbol{v}_\theta \in \mathcal{V}$ such that $f_{\boldsymbol{v}_\theta}(\theta') = \mathbf{1}_{\{\theta\}}(\theta')$. In other words, $f_{\boldsymbol{v}_\theta}$ is a discrete distribution that is concentrated at θ. Given $\mathcal{V}$ and f, we rewrite the original problem (1) where we search over policies into the following problem which searches over policy distributions:

$$\boldsymbol{v}^* = \operatorname*{arg\,max}_{\boldsymbol{v} \in \mathcal{V}} \mathbb{E}_{\theta \sim f_{\boldsymbol{v}}}[G_J(\pi_\theta) \mid \pi_\theta \in \Pi_{Z,d}]. \tag{2}$$

We show the connection between (1) and (2) with Lemma 2.

Lemma 2. *Let π_{θ^*} and v^* be any solution to* (1) *and* (2) *respectively. Then $G_J(\pi_{\theta^*}) = \mathbb{E}_{\theta \sim f_{\boldsymbol{v}^*}}[G_J(\pi_\theta) \mid \pi_\theta \in \Pi_{Z,d}]$.*

Proof. If π_{θ^*} is a solution to (1), then $\pi_{\theta^*} \in \Pi_{Z,d}$ and $G_J(\pi_{\theta^*}) \geq G_J(\pi_\theta)$ for all $\pi_\theta \in \Pi_{Z,d}$. Therefore,

$$\begin{aligned} G_J(\pi_{\theta^*}) &= \mathbb{E}_{\theta \sim f_{\boldsymbol{v}_{\theta^*}}}[G_J(\pi_\theta) \mid \pi_\theta \in \Pi_{Z,d}] \\ &\leq \mathbb{E}_{\theta \sim f_{\boldsymbol{v}^*}}[G_J(\pi_\theta) \mid \pi_\theta \in \Pi_{Z,d}] \\ &\leq \mathbb{E}_{\theta \sim f_{\boldsymbol{v}^*}}[G_J(\pi_{\theta^*})] = G_J(\pi_{\theta^*}), \end{aligned}$$

where the first inequality holds since $\boldsymbol{v}^*$ is a solution to (2). □

B. Surrogate Objective

As with other CE-based algorithms, we replace the objective in (2) with a surrogate function. For the unconstrained CE method, the surrogate function is the conditional expectation of G_J over the elite sample policies with the current sampling distribution $f_{\boldsymbol{v}}$. The ranking function for unconstrained CE is defined using the concept of ρ-quantiles for random variables, which is formally defined as below.

Definition 1. *[42] Given a distribution $P \in \mathcal{D}(\mathbb{R})$, $\rho \in (0,1)$ and a random variable $X \sim P$, the ρ-quantile of X is defined as a scalar γ such that $Pr(X \leq \gamma) \geq \rho$ and $Pr(X \geq \gamma) \geq 1 - \rho$.*

For $\rho \in (0,1)$, $\boldsymbol{v} \in \mathcal{V}$ and any function $X : \Theta \to \mathbb{R}$, we denote the ρ-quantile of X for $\theta \sim f_{\boldsymbol{v}}$ by $\xi_X(\rho, \boldsymbol{v})$. Let $\delta : \mathbb{R} \times \{\geq, \leq, >, <, =\} \times \mathbb{R} \to \{0,1\}$ be an indicator function such that for $\circ \in \{\geq, \leq, >, <, =\}$, $\delta(x \circ y) = 1$ if and only if $x \circ y$ holds. Usually, we interpret ρ as the proportion of highly ranked policies. For the unconstrained CE method, a policy π_θ is considered as highly ranked if $G(\pi_\theta) \geq \xi_G(1 - \rho, \boldsymbol{v})$, that is, if the G-value of π_θ is greater than at least $(1 - \rho)$ of all policies in Π_Θ with sampling distribution $f_{\boldsymbol{v}}$. The surrogate objective function for the unconstrained CE method is

$$\mathbb{E}_{\theta \sim f_{\boldsymbol{v}}}[G(\pi_\theta)\delta(G(\pi_\theta) \geq \xi_G(1 - \rho, \boldsymbol{v}))]. \tag{3}$$

When there is a constraint $H(\pi) \leq d$, we also need to take the H-value of π_θ into consideration while designing ranking functions. As in the unconstrained case, we will have a ρ proportion of all policies as highly ranked policies. Let $p_{\boldsymbol{v}}$ be the probability of sampling feasible policies with $f_{\boldsymbol{v}}$. The definition of highly-ranked policies with respect to $f_{\boldsymbol{v}}$ can be split into two cases, depending whether $p_{\boldsymbol{v}} \geq \rho$ or not.

a) Case 1: If $p_{\boldsymbol{v}} < \rho$, the ρ-quantile of H with distribution $f_{\boldsymbol{v}}$ will be greater than the constraint threshold d. In this case, we rank policies in the decreasing order of their H-values. The indicator function of highly ranked policies is $\delta(H(\pi_\theta) \leq \xi_H(\rho, \boldsymbol{v}))$. As a result, all feasible policies and a small proportion (to be specific, $(\rho - p_{\boldsymbol{v}}) \backslash (1 - p_{\boldsymbol{v}})$) of infeasible policies with the least H-values will be highly ranked.

b) Case 2: If $p_{\boldsymbol{v}} \geq \rho$, the probability of sampling feasible policies with $f_{\boldsymbol{v}}$ is at least ρ. In this case, we rank *feasible* policies in the increasing order of their G-values. Define $U : \Pi_\Theta \to \mathbb{R}$ such that $U(\pi_\theta) := G(\pi_\theta)\delta(H(\pi_\theta) \leq d)$. The indicator function of highly ranked policies is $\delta(U(\pi_\theta) \geq \xi_U(1-\rho, \boldsymbol{v}))$. Since G_J is strictly positive, $U(\pi) > U(\pi')$ holds for any feasible π and infeasible π'. As $p_{\boldsymbol{v}} \geq \rho$, any policy π_θ such that $U(\pi_\theta) \geq \xi_U(1-\rho, \boldsymbol{v}))$ will be feasible. As a result, a fraction of $\rho \backslash p_{\boldsymbol{v}}$ feasible policies with the highest G-values will be highly ranked.

We can combine the two cases and write down a single indicator function of highly ranked policies with distribution $f_{\boldsymbol{v}}$. Define $S : \Pi_\Theta \times \mathcal{V} \times (0,1) \to \{0,1\}$ such that

$$\begin{aligned} S(\pi_\theta, \boldsymbol{v}, \rho) :=& \delta(\xi_H(\rho, \boldsymbol{v}) > d)\delta(H(\pi_\theta) \leq \xi_H(\rho, \boldsymbol{v})) + \\ & \delta(\xi_H(\rho, \boldsymbol{v}) \leq d)\delta(U(\pi_\theta) \geq \xi_U(1-\rho, \boldsymbol{v})). \end{aligned}$$

Then the surrogate function for CCE can be expressed as follows:

$$L(\boldsymbol{v}; \rho) = \mathbb{E}_{\theta \sim f_{\boldsymbol{v}}}[G(\pi_\theta)S(\pi_\theta, \boldsymbol{v}, \rho)]. \tag{4}$$

Note that the surrogate function (4) for the constrained problem has the same structure as that for the unconstrained problem in (3). Intuitively, the highly-ranked policies are selected to update the current policy distribution $f_{\boldsymbol{v}}$. If $p_{\boldsymbol{v}} < \rho$, it suggests that the distribution update should be focused on increasing the probability to sample feasible policies; if $p_{\boldsymbol{v}} \geq \rho$, we can pay more attention to increasing the expected G-value over feasible policies.

Remark 1. *For the unconstrained problem, $p_{\boldsymbol{v}} = 1 > \rho$ and $U(\pi_\theta) = G(\pi_\theta)$, then*

$$\begin{aligned} & \mathbb{E}_{\theta \sim f_{\boldsymbol{v}}}[\delta(G(\pi_\theta) \geq \xi_G(1-\rho, \boldsymbol{v}))] \\ &= \mathbb{E}_{\theta \sim f_{\boldsymbol{v}}}[G(\pi_\theta)\delta(U(\pi_\theta) \geq \xi_U(1-\rho, \boldsymbol{v}))] = L(\boldsymbol{v}; \rho). \end{aligned}$$

Therefore (3) *is a special case of* (4).

Remark 2. *If $\xi_H(\rho, \boldsymbol{v}) \leq d$, then*

$$\begin{aligned} & G(\pi_\theta)\delta(G(\pi_\theta) \geq \xi_G(1-\rho, \boldsymbol{v})) \\ \geq\ & U(\pi_\theta)\delta(U(\pi_\theta) \geq \xi_U(1-\rho, \boldsymbol{v})) \\ \geq\ & G(\pi_\theta)\delta(H(\pi_\theta) \leq \xi_H(\rho, \boldsymbol{v})). \end{aligned}$$

Intuitively, if at least $100\rho\%$ of all policies are feasible, $L(\boldsymbol{v}; \rho)$ is less than the objective value for the unconstrained CE method and greater than the expected G-value over the $100\rho\%$ policies of the lowest H-values.

Remark 3. *For ease of analysis, we may approximate δ by a continuous function $\tilde{\delta}_\varepsilon$ where $\varepsilon > 0$, such that for any $x, y \in \mathbb{R}$ and $\circ \in \{\geq, >\}$:*

$$\tilde{\delta}_\varepsilon(x \circ y) = \begin{cases} \delta(x \circ y) & \textit{if } y \circ x \textit{ or } y < x - \varepsilon \\ (y-x)/\varepsilon + 1 & \textit{otherwise.} \end{cases}$$

$$\tilde{\delta}_\varepsilon(x < y) = 1 - \tilde{\delta}_\varepsilon(x \geq y),\ \tilde{\delta}_\varepsilon(x \leq y) = 1 - \tilde{\delta}_\varepsilon(x > y).$$

The main problem we solve in this paper can be then stated as follows.

Problem 1. *Given a set $\Pi = \{\pi_\theta : \theta \in \Theta\}$ of policies with parameter space Θ, a set $F_{\mathcal{V}} = \{f_{\boldsymbol{v}} \in \mathcal{D}(\Theta) : \boldsymbol{v} \in \mathcal{V}\}$ of distributions over Θ, two functions $G : \Pi \to \mathbb{R}^+$ and $H : \Pi \to \mathbb{R}$, a constraint upper bound d and $\rho \in (0,1)$, compute $\boldsymbol{v}^* \in \mathcal{V}$ such that*

$$\boldsymbol{v}^* = \arg\max_{\boldsymbol{v} \in \mathcal{V}} L(\boldsymbol{v}; \rho),$$

where $L : \mathcal{V} \times (0,1) \to \mathbb{R}$ is defined in (4).

C. The Constrained Cross-Entropy Algorithm

In this section, we focus on how to solve Problem 1 and propose the CCE algorithm. We first describe the key idea behind the (idealized) CE-based stochastic optimization method as in [43]. For notational simplicity, we use $\mathbb{E}_{\boldsymbol{v}}[\cdot]$ to represent $\mathbb{E}_{\theta \sim f_{\boldsymbol{v}}}[\cdot]$ in the rest of this paper.

As explained in the previous section, we aim at finding a policy distribution $f_{\boldsymbol{v}^*}$ to maximize $L(\boldsymbol{v}; \rho)$. By definition of ρ-quantiles, it is a rare event to sample the highly ranked policies for small ρ. The idea behind CE is to treat this optimization problem as an estimation problem of rare-event probabilities. With importance sampling, we may estimate $L(\boldsymbol{v}; \rho)$ using any sampling distribution g that shares the same support Θ as $f_{\boldsymbol{v}}$, then $L(\boldsymbol{v}; \rho) = \mathbb{E}_g[G(\pi_\theta)S(\pi_\theta, \boldsymbol{v}, \rho)\frac{f_{\boldsymbol{v}}(\theta)}{g(\theta)}]$. It is well-known that the optimal distribution $g^*_{\boldsymbol{v}}$ [44] with minimal variance is

$$g^*_{\boldsymbol{v}}(\theta) = \frac{G(\pi_\theta)S(\pi_\theta, \boldsymbol{v}, \rho)f_{\boldsymbol{v}}(\theta)}{L(\boldsymbol{v}; \rho)}. \tag{5}$$

In practice we smoothen the updates by including a learning rate $\alpha \in (0,1)$ so the goal distribution is $\tilde{g}_{\boldsymbol{v}} = \alpha g^*_{\boldsymbol{v}} + (1-\alpha) f_{\boldsymbol{v}}$. Since neither $g^*_{\boldsymbol{v}}$ nor $\tilde{g}_{\boldsymbol{v}}$ are necessarily in $F_{\mathcal{V}}$, we project $\tilde{g}_{\boldsymbol{v}}$ to $f_{\boldsymbol{v}'} \in F_{\mathcal{V}}$ by minimizing the Kullback-Leibler (KL) divergence [45] between $f_{\boldsymbol{v}''} \in F_{\mathcal{V}}$ and $\tilde{g}_{\boldsymbol{v}}$, which is also equivalent to minimizing the cross entropy between $\tilde{g}_{\boldsymbol{v}}$ and $f_{\boldsymbol{v}''}$.

$$\begin{aligned} \boldsymbol{v}' &= \arg\min_{\boldsymbol{v}'' \in \mathcal{V}} D_{KL}(\tilde{g}_{\boldsymbol{v}} \mid\mid f_{\boldsymbol{v}''}) \\ &= \arg\max_{\boldsymbol{v}'' \in \mathcal{V}} \mathbb{E}_{\theta \sim \tilde{g}_{\boldsymbol{v}}}[\log f_{\boldsymbol{v}''}(\theta)] \\ &= \arg\max_{\boldsymbol{v}'' \in \mathcal{V}} \Big(\alpha \mathbb{E}_{\boldsymbol{v}}\big[\frac{G(\pi_\theta)S(\pi_\theta, \boldsymbol{v}, \rho)}{L(\boldsymbol{v}; \rho)} \log f_{\boldsymbol{v}''}(\theta)\big] \\ &\quad + (1-\alpha)\mathbb{E}_{\boldsymbol{v}}\big[\log f_{\boldsymbol{v}''}(\theta)\big]\Big). \end{aligned} \tag{6}$$

We focus ourselves on a specific family of distributions over Θ called *natural exponential family* (NEF), which includes many commonly used distributions such as Gaussian distribution and Gamma distribution. NEF is defined as follows.

Definition 2. *A parameterized family $F_{\mathcal{V}} = \{f_{\boldsymbol{v}} \in \mathcal{D}(\Theta), \boldsymbol{v} \in \mathcal{V} \subseteq \mathbb{R}^{d_v}\}$ is called a* natural exponential family *if there exist continuous mappings $\Gamma : \mathbb{R}^{d_\theta} \to \mathbb{R}^{d_v}$ and $K : \mathbb{R}^{d_\theta} \to \mathbb{R}$ such that $f_{\boldsymbol{v}}(\theta) = \exp\big(\boldsymbol{v}^\intercal \Gamma(\theta) - K(\boldsymbol{v})\big)$, where $\mathcal{V} \subseteq \{\boldsymbol{v} \in \mathbb{R}^{d_v} : |K(\boldsymbol{v})| < \infty\}$ is the natural parameter space and $K(\boldsymbol{v}) = \log \int_\Theta \exp\big(\boldsymbol{v}^\intercal \Gamma(\theta)\big) d\theta$.*

Define $m(\boldsymbol{v}) := \mathbb{E}_{\boldsymbol{v}}[\Gamma(\theta)] \in \mathbb{R}^{d_v}$ for $\boldsymbol{v} \in \mathcal{V}$, which is continuously differentiable in $\boldsymbol{v}$. It can be verified that $m(\boldsymbol{v}) = \frac{\partial}{\partial \boldsymbol{v}} K(\boldsymbol{v})$ and $\frac{\partial}{\partial \boldsymbol{v}} m(\boldsymbol{v}) = \mathrm{Cov}_{\boldsymbol{v}}[\Gamma(\theta)]$ where $\mathrm{Cov}_{\boldsymbol{v}}[\Gamma(\theta)]$ denotes the covariance matrix of $\Gamma(\theta)$ with $\theta \sim f_{\boldsymbol{v}}$. We take Assumption 1 to guarantee that m^{-1} exists and is continuously differentiable over $\{\eta : \exists\ \boldsymbol{v} \in int(\mathcal{V})\ s.t.\ \eta = m(\boldsymbol{v})\}$. The proof can be done by directly applying the inverse function theorem to m on $int(\mathcal{V})$.

Assumption 1. $\mathrm{Cov}_{\boldsymbol{v}}[\Gamma(\theta)]$ *is positive definite for any* $\boldsymbol{v} \in \mathcal{V} \subseteq int(\{\boldsymbol{v} \in \mathbb{R}^{d_v} : |K(\boldsymbol{v})| < \infty\})$.

With Assumption 1, $\nabla^2 K(\boldsymbol{v}) = \mathrm{Cov}_{\boldsymbol{v}}[\Gamma(\theta)] \succcurlyeq 0$ and thus $K(\boldsymbol{v})$ is convex in $\boldsymbol{v}$. Thus $\log f_{\boldsymbol{v}''}(\theta) = (\boldsymbol{v}'')^\intercal \Gamma(\theta) - K(\boldsymbol{v}'')$ is concave in $\boldsymbol{v}''$. As a result, $\boldsymbol{v}'$ in (6) can be found by setting $\frac{\partial}{\partial \boldsymbol{v}''}\Big(- \int_\Theta \tilde{g}_{\boldsymbol{v}}(\theta) \log f_{\boldsymbol{v}''}(\theta) d\theta\Big) = \mathbf{0}$, which induces

$$m(\boldsymbol{v}') - m(\boldsymbol{v}) = \alpha\big(\mathbb{E}_{g_{\boldsymbol{v}}^*}[\Gamma(\theta)] - m(\boldsymbol{v})\big). \tag{7}$$

As a property of NEF, the KL-divergence of $f_{\boldsymbol{v}}$ from g satisfies $\frac{\partial}{\partial \boldsymbol{v}} D_{KL}(g \mid\mid f_{\boldsymbol{v}}) = -\mathbb{E}_g[\Gamma(\theta)] + m(\boldsymbol{v})$. Therefore

$$m(\boldsymbol{v}') - m(\boldsymbol{v}) = -\alpha\Big(\frac{\partial}{\partial \boldsymbol{v}''} D_{KL}(g_{\boldsymbol{v}}^* \mid\mid f_{\boldsymbol{v}''})\Big)\Big|_{\boldsymbol{v}''=\boldsymbol{v}}, \tag{8}$$

which shows that if $\boldsymbol{v}$ is updated to $\boldsymbol{v}'$ by solving (6), $m(\boldsymbol{v})$ will be updated in the negative gradient direction of the objective function $D_{KL}(g_{\boldsymbol{v}}^* \mid\mid f_{\boldsymbol{v}})$ where $g_{\boldsymbol{v}}^*$ is the optimal sampling distribution from importance sampling.

Define $\tilde{L}(\boldsymbol{v};\rho) := \mathbb{E}_{g_{\boldsymbol{v}}^*}[\Gamma(\theta)] - m(\boldsymbol{v})$. If G is bounded with a strictly positive lower bound, then

$$\begin{aligned}
\tilde{L}(\boldsymbol{v};\rho) &= \frac{\mathbb{E}_{\boldsymbol{v}}[G(\pi_\theta)S(\pi_\theta,\boldsymbol{v},\rho)\Gamma(\theta)]}{L(\boldsymbol{v};\rho)} - m(\boldsymbol{v}) \\
&= \int_\Theta \frac{G(\pi_\theta)S(\pi_\theta,\boldsymbol{v},\rho)}{L(\boldsymbol{v};\rho)} f_{\boldsymbol{v}}(\theta)(\Gamma(\theta) - m(\boldsymbol{v}))d\theta \\
&\overset{(*)}{=} \int_\Theta \frac{G(\pi_\theta)S(\pi_\theta,\boldsymbol{v},\rho)}{L(\boldsymbol{v};\rho)} \big(\frac{\partial}{\partial \boldsymbol{v}} f_{\boldsymbol{v}}(\theta)\big) d\theta \\
&\overset{(**)}{=} \frac{\partial}{\partial \boldsymbol{v}''} \frac{\mathbb{E}_{\boldsymbol{v}''}[G(\pi_\theta)S(\pi_\theta,\boldsymbol{v},\rho)]}{L(\boldsymbol{v};\rho)}\Big|_{\boldsymbol{v}''=\boldsymbol{v}} \\
&= \frac{\partial}{\partial \boldsymbol{v}''} \log \mathbb{E}_{\boldsymbol{v}''}[G(\pi_\theta)S(\pi_\theta,\boldsymbol{v},\rho)]\Big|_{\boldsymbol{v}''=\boldsymbol{v}},
\end{aligned} \tag{9}$$

where the $(*)$ step holds by noticing $\frac{\partial}{\partial \boldsymbol{v}} f_{\boldsymbol{v}}(\theta) = f_{\boldsymbol{v}}(\theta)(\Gamma(\theta) - m(\boldsymbol{v}))$ and the $(**)$ step holds by the dominated convergence theorem. Combining (7) and (9), we get

$$\begin{aligned}
&m(\boldsymbol{v}') - m(\boldsymbol{v}) = \alpha \tilde{L}(\boldsymbol{v};\rho) \\
&= \alpha \frac{\partial}{\partial \boldsymbol{v}''} \log \mathbb{E}_{\boldsymbol{v}''}[G(\pi_\theta)S(\pi_\theta,\boldsymbol{v},\rho)]\Big|_{\boldsymbol{v}''=\boldsymbol{v}},
\end{aligned} \tag{10}$$

which leads to the second interpretation of the updates: The update from $\boldsymbol{v}$ to $\boldsymbol{v}'$ approximately follows the gradient direction of $\log L(\boldsymbol{v}'';\rho)$, while the quantiles are estimated using the previous distribution $f_{\boldsymbol{v}}$.

If we apply $\log f_{\boldsymbol{v}''}(\theta) = (\boldsymbol{v}'')^\intercal \Gamma(\theta) - K(\boldsymbol{v}'')$ to (6), we can simplify the right-hand side (RHS) of (6) as

$$\big(\alpha \mathbb{E}_{\boldsymbol{v}}\big[\frac{G(\pi_\theta)S(\pi_\theta,\boldsymbol{v},\rho)\Gamma(\theta)}{L(\boldsymbol{v};\rho)}\big] + (1-\alpha) m(\boldsymbol{v})\big)^\intercal \boldsymbol{v}'' - K(\boldsymbol{v}''),$$

which is concave in $\boldsymbol{v}''$. By setting the derivative with respect to $\boldsymbol{v}''$ as zero, we get an explicit expression of $m(\boldsymbol{v}')$ as in (11).

$$m(\boldsymbol{v}') = \alpha \mathbb{E}_{\boldsymbol{v}}\big[\frac{G(\pi_\theta)S(\pi_\theta,\boldsymbol{v},\rho)\Gamma(\theta)}{L(\boldsymbol{v};\rho)}\big] + (1-\alpha) m(\boldsymbol{v}). \tag{11}$$

The pseudocode of the CCE algorithm is given in Algorithm 1, which approximately takes the updates in (11) in each iteration, with all expectations and quantiles estimated by Monte Carlo simulation. Given $f_{\boldsymbol{v}_l} \in \mathcal{D}(\Theta)$ in the l^{th} iteration, we sample over policies (Step 3), evaluate their G-values and H-values (Step 5), estimate $S(\cdot,\boldsymbol{v},\rho)$ (Step 7 to 10) and estimate $m(\boldsymbol{v}_{l+1})$ with $\hat{\eta}_{l+1}$ (Step 11) and finally update the sampling distribution to $\boldsymbol{v}_{l+1}$ (Step 12).

Algorithm 1 Constrained Cross-Entropy Method

```
Require: An objective function $G$, a constraint function $H$, a constraint upper bound $d$, a class of parameterized policies $\Pi_\Theta$, an NEF family $F_\mathcal{V}$.
 1: $l \leftarrow 1$. Initialize $n_l, \boldsymbol{v}_l, \rho, \lambda_l, \alpha_l$. $k_l \leftarrow \lceil \rho n_l \rceil$. $\hat{\eta}_l \leftarrow \mathbf{0}$.
 2: repeat
 3:     Sample $\theta_1, \ldots, \theta_{n_l} \sim f_{\boldsymbol{v}_l}$ i.i.d..
 4:     for $i = 1, \ldots, n_l$ do
 5:         Simulate $\pi_{\theta_i}$ and estimate $G(\pi_{\theta_i})$, $H(\pi_{\theta_i})$.
 6:     end for
 7:     Sort $\{\theta_i\}_{i=1}^{n_l}$ in ascending order of $H$. Let $\Lambda_l$ be the first $k_l$ elements.
 8:     if $H(\pi_{\theta_{k_l}}) \leq d$ then
 9:         Sort $\{\theta_i \mid H(\pi_{\theta_i}) \leq d\}$ in descending order of $G$. Let $\Lambda_l$ be the first $k_l$ elements.
10:     end if
11:     $\hat{\eta}_{l+1} \leftarrow \alpha_l \sum_{\theta \in \Lambda_l} \frac{G(\pi_\theta)}{\sum_{\theta \in \Lambda_l} G(\pi_\theta)} \Gamma(\theta) + (1 - \alpha_l)\big(\frac{\lambda_l}{n_l} \sum_{i=1}^{n_l} \Gamma(\theta_i) + (1-\lambda_l)\hat{\eta}_l\big)$.
12:     $\boldsymbol{v}_{l+1} \leftarrow m^{-1}(\hat{\eta}_{l+1})$.
13:     Update $n_l, \lambda_l, \alpha_l$. $l \leftarrow l+1$. $k_l \leftarrow \lceil \rho n_l \rceil$.
14: until The maximum number of iterations is reached.
```

D. Convergence Analysis

We prove the convergence of Algorithm 1 by comparing the asymptotic behavior of $\{\hat{\eta}_l\}_{l \geq 0}$ with the flow induced by the following ordinary differential equation (ODE):

$$\frac{\partial \eta(t)}{\partial t} = \tilde{L}(m^{-1}(\eta(t));\rho), \tag{12}$$

where we define $\eta := m(\boldsymbol{v})$ or equivalently, $\boldsymbol{v} = m^{-1}(\eta)$. The main result that connects the asymptotic behavior of Algorithm 1 with that of an ODE is stated in Theorem 1.

Theorem 1. *If Assumptions 1 and 2 hold, the sequence* $\{\hat{\eta}_l\}_{l \geq 0}$ *in Step 11 of Algorithm 1 converges to a connected internally chain recurrent set of* (12) *as* $l \to \infty$ *with probability 1.*

By definition of η in (12), we know $\frac{\partial \eta(t)}{\partial t} = \frac{\partial \boldsymbol{v}}{\partial t} \cdot \mathrm{Cov}_{\boldsymbol{v}}[\Gamma(\theta)]$. Since $\mathrm{Cov}_{\boldsymbol{v}}[\Gamma(\theta)]$ is invertible by Assumption 1, (12) can be rewritten with variable $\boldsymbol{v}$

$$\frac{\partial \boldsymbol{v}}{\partial t} = \Big(\tilde{L}(\boldsymbol{v};\rho)\Big)^\intercal \big(\mathrm{Cov}_{\boldsymbol{v}}[\Gamma(\theta)]\big)^{-1}. \tag{13}$$

The conclusion of Theorem 1 can be equivalently stated in terms of the variable $\boldsymbol{v}$: the sequence $\{\boldsymbol{v}_l\}_{l \geq 0}$ of Algorithm 1 converges to a connected internally chain recurrent set of (13) as $l \to \infty$ with probability 1.

Intuitively, a point $\boldsymbol{v}_0 \in \mathcal{V}$ is *chain recurrent* for (13) if the solution $\boldsymbol{v}(t)$ of (13) with initial condition $\boldsymbol{v}(0) = \boldsymbol{v}_0$ can return to $\boldsymbol{v}_0$ within some finite time $t' > 0$ itself or just with finitely many arbitrarily small perturbations. An *internally* *chain recurrent set* is a nonempty compact *invariant* set of chain-recurrent points. In other words, $\boldsymbol{v}$ can never leave an internally chain recurrent set if $\boldsymbol{v}_0$ belongs to it.

Theorem 1 implies that with probability 1, the set of points that occur infinitely often in $\{\boldsymbol{v}_l\}_{l\geq 0}$ are internally chain recurrent for (13). Since $f_{\boldsymbol{v}}$ belongs to NEF, $\mathrm{Cov}_{\boldsymbol{v}}[\Gamma(\theta)]$ is the Fisher information matrix at $\boldsymbol{v}$ and the right hand side of (13) is an estimate of the natural gradient of $\log L(\boldsymbol{v};p)$ with a fixed indicator function S. This suggests that $\boldsymbol{v}$ evolves to increase $L(\boldsymbol{v};\rho)$, which is consistent with the optimization problem (4) and our motivation to solve a constrained RL problem. Note that internally chain-recurrent sets are generally not unique and our algorithm can still converge to a local optimum.

We need a series of assumptions for technical reasons.

Assumption 2. *(2a)* $\tilde{L}(\boldsymbol{v};\rho)$ *is continuous in* $\boldsymbol{v}\in int(\mathcal{V})$ *and* (12) *has a unique integral curve for any given initial condition.*

(2b) The number of samples in the l^{th} *iteration is* $n_l=\Theta(l^\beta)$, $\beta>0$. *The gain sequence* $\{\alpha_l\}$ *is positive and decreasing with* $\lim_{l\to\infty}\alpha_l=0$, $\sum_{l=1}^{\infty}\alpha_l=\infty$. $\{\lambda_l\}$ *satisfies* $\lambda_l=O(\frac{1}{l^\lambda})$ *for some* $\lambda>0$ *such that* $\beta+2\lambda>1$.

(2c) For any $\rho\in(0,1)$ *and* $f_{\boldsymbol{v}}$ *for any* $\boldsymbol{v}\in\mathcal{V}$, *the* ρ*-quantile of* $\{H(\pi_\theta):\theta\sim f_{\boldsymbol{v}}\}$ *and the* $(1-\rho)$*-quantile of* $\{U(\pi_\theta):\theta\sim f_{\boldsymbol{v}}\}$ *are both unique.*

(2d) Both Θ *and* $\mathcal{V}$ *are compact.*

(2e) The function G *defined in Problem 1 is bounded and has a positive lower bound:* $\inf_{\pi\in\Pi}G(\pi)>0$. *The function* H *in Problem 1 is bounded.*

(2f) $\boldsymbol{v}_l\in int(\mathcal{V})$ *for any iteration* l.

Assumption (2a) ensures that (12) is well-posed and has a unique solution. Assumption (2b) addresses some requirements on the number of sampled policies in each iteration and other hyperparameters in Algorithm 1. Assumptions (2c) to (2e) are used in the proof of the convergence of Algorithm 1. Assumption (2c) is required to show that $\frac{1}{n_l}\sum_{\theta\in\Lambda_l}G(\pi_\theta)$ in Step 11 of Algorithm 1 is an unbiased estimate of $\mathbb{E}_{\boldsymbol{v}_l}[G(\pi_\theta)S(\pi_\theta,\boldsymbol{v}_l,\rho)]$. Assumption (2d) and (2e) are compactness and boundedness constraints for the sets and functions involved in Algorithm 1, which are unlikely to be restrictive in practice. Assumption (2f) states that $\mathcal{V}$ is large enough such that the learned $\boldsymbol{v}$ lies within its interior.

The main idea behind the proof of Theorem 1 is similar to that of Theorem 3.1 in [43], although the details are tailored to our problem. There are two major parts in the convergence proof: The first part shows that all the sampling-based estimates converge to the true values almost surely, including sample quantiles and sample estimates of G, H and L. The second part shows that the asymptotic behavior of the idealized updates in (7) can be described by the ODE (12).

In practice we can only estimate the expectations and quantiles in (11) using finite samples. Let $\mathcal{Y}_l=\{\theta_1,\ldots,\theta_{n_l}\}$ be the set of samples in the l^{th} iteration with sampling distribution $f_{\boldsymbol{v}_l}$. We denote the sample estimate of $S(\pi_\theta,\boldsymbol{v},\rho)$ as $\hat{S}(\pi_\theta,\boldsymbol{v},\rho)$.

Consider the equation in the Step 11 of Algorithm 1:

$$\begin{aligned}\hat{\eta}_{l+1}=&\alpha_l\frac{\sum_{i=1}^{n_l}G(\pi_{\theta_i})\hat{S}(\pi_{\theta_i},\boldsymbol{v}_l,\rho)\Gamma(\theta_i)}{\sum_{i=1}^{n_l}G(\pi_{\theta_i})\hat{S}(\pi_{\theta_i},\boldsymbol{v}_l,\rho)}\\&+(1-\alpha_l)\Big(\frac{\lambda_l}{n_l}\sum_{i=1}^{n_l}\Gamma(\theta_i)+(1-\lambda_l)\hat{\eta}_l\Big),\end{aligned}\tag{14}$$

where $\boldsymbol{v}_l=m^{-1}(\hat{\eta}_l)$ and $\boldsymbol{v}_{l+1}=m^{-1}(\hat{\eta}_{l+1})$. We can rewrite (14) as

$$m(\boldsymbol{v}_{l+1})-m(\boldsymbol{v}_l)=\hat{\eta}_{l+1}-\hat{\eta}_l=\alpha_l\Big(\tilde{L}(\boldsymbol{v}_l;\rho)+b_l+w_l\Big),\tag{15}$$

where

$$\begin{aligned}b_l=&\frac{\sum_{i=1}^{n_l}G(\pi_{\theta_i})\hat{S}(\pi_{\theta_i},\boldsymbol{v}_l,\rho)\Gamma(\theta_i)}{\sum_{i=1}^{n_l}G(\pi_{\theta_i})\hat{S}(\pi_{\theta_i},\boldsymbol{v}_l,\rho)}\\&-\frac{\mathbb{E}_{\boldsymbol{v}_l}[G(\pi_\theta)S(\pi_\theta,\boldsymbol{v}_l,\rho)\Gamma(\theta)]}{\mathbb{E}_{\boldsymbol{v}_l}[G(\pi_\theta)S(\pi_\theta,\boldsymbol{v}_l,\rho)]},\\w_l=&\frac{1-\alpha_l}{\alpha_l}\Big(\frac{\lambda_l}{n_l}\sum_{i=1}^{n_l}\Gamma(\theta_i)-\lambda_l\hat{\eta}_l\Big).\end{aligned}\tag{16}$$

Comparing (15) and (10), we see that the error is sampling-based estimation of all expectations and quantiles is captured by b_l and w_l.

We aim to show the connection between $\{\hat{\eta}_l\}_{l\geq 0}$ and the ODE (12) using the following conclusion in stochastic approximation.

Theorem 2. *(Theorem 1.2, [46], with modified notation) Let* $Y:\mathbb{R}^m\to\mathbb{R}^m$ *be a continuous vectorfield with unique integral curves. Let* $\{v_n\}_{n\geq 0}$ *be the solution to* $v_{n+1}-v_n=\gamma_n(Y(v_n)+u_n+b_n)$, *where* $\{\gamma_n\}_{n\geq 0}$ *is a decreasing gain sequence. Assume that*

- $\{\gamma_n\}_{n\geq 0}$ *is bounded.*
- $\lim_{n\to+\infty}b_n=0$.
- *For any* $N>0$,

$$\lim_{n\to\infty}\Big(\sup_{k:0\leq\tau_k-\tau_n\leq N}||\sum_{i=n}^{k-1}\gamma_i u_i||\Big)=0,$$

where $\{\tau_n\}_{n\in\mathbb{N}}$ *is defined as:* $\tau_0=0$, $\tau_n=\sum_{i=0}^{n-1}\gamma_i$.

Then the limit set of $\{v_n\}_{n\geq 0}$ *is a connected set internally chain-recurrent for the flow induced by* Y.

We first show that $\lim_{l\to\infty}b_l=0$ where b_l is defined in (16), which is stated in Lemma 3.

Lemma 3. *With Assumption (2b), (2c), (2d), (2e),* $\lim_{l\to\infty}b_l=0$, *with probability 1.*

In order to prove Lemma 3, we first show that the sample quantile is an unbiased estimate of the true quantile, which is stated in Lemma 4. Although we only show the result for the ρ-quantile of H, similar results apply for the $(1-\rho)$-quantile of U.

Lemma 4. *Given* $\rho\in(0,1)$, *let* $\xi(\rho,\boldsymbol{v}_l)$ *be the true* ρ*-quantile of* $H(\pi_\theta)$ *with* $\theta\sim f_{\boldsymbol{v}_l}$ *and* $\hat{\xi}_l$ *be a sample* ρ*-quantile acquired from* n_l *i.i.d. samples. With Assumption (2b), (2c), (2d), (2e),* $\hat{\xi}_l-\xi(\rho,\boldsymbol{v}_l)\to 0$ *as* $l\to\infty$ *with probability 1.*

Proof. By Assumption (2e), $H(\pi_\theta) \in \mathcal{H} := [H_{min}, H_{max}]$ for all $\pi_\theta \in \Pi_\Theta$ for some $H_{min}, H_{max} \in \mathbb{R}$. It can be verified that any ρ-quantile $\xi(\rho, \boldsymbol{v}_l)$ with $\theta \sim f_{\boldsymbol{v}_l}(\cdot)$ can be represented as an optimal solution of the following optimization problem [42]:

$$\min_{\gamma \in \mathcal{H}} J_l(\gamma) := \mathbb{E}_{\boldsymbol{v}_l}[h(H(\pi_\theta), \gamma)]$$

$$s.t.\ h(H(\pi_\theta), \gamma) = \begin{cases} \rho(H(\pi_\theta) - \gamma), & \text{if } H(\pi_\theta) \geq \gamma, \\ (1-\rho)(\gamma - H(\pi_\theta)), & \text{if } H(\pi_\theta) < \gamma. \end{cases}$$

Similarly the sample ρ-quantile $\hat{\xi}_l$ can be computed by minimizing

$$\hat{J}_l(\gamma) := \frac{1}{n_l} \sum_{i=1}^{n_l} h(H(\pi_{\theta_i}), \gamma),$$

where $\{\theta_1, \ldots, \theta_{n_l}\}$ are i.i.d. samples with distribution $f_{\boldsymbol{v}_l}$.

We first show that $J_l(\gamma)$ uniformly converges to $\hat{J}_l(\gamma)$ over $\mathcal{H}$ with probability 1, i.e. $\sup_{\gamma \in \mathcal{H}} |J_l(\gamma) - \hat{J}_l(\gamma)| \to 0$ as $l \to \infty$ with probability 1.

Let δ and r be two arbitrary scalars such that $\delta > 0$ and $r \leq \frac{\delta}{3\max(\rho, 1-\rho)}$. Let $B(\gamma, r) := \{\gamma' \in \mathcal{H} : ||\gamma - \gamma'|| \leq r\}$ be the r-neighborhood of $\gamma \in \mathcal{H}$ within $\mathcal{H}$. Since $\mathcal{H}$ is compact, there exists a finite set $\mathcal{U} = \{h_1, \ldots, h_k\} \subset \mathcal{H}$ such that $\mathcal{H} \subseteq \bigcup_{i=1}^k B(h_i, r)$. For each $\gamma \in \mathcal{H}$, let $u(\gamma) \in \mathcal{U}$ be the closest component in $\mathcal{U}$. By definition, $\sup_{\gamma \in \mathcal{H}} ||\gamma - u(\gamma)|| \leq r$. For any $\gamma \in \mathcal{H}$,

$$\begin{aligned} & |J_l(\gamma) - J_l(u(\gamma))| \\ = & |\mathbb{E}_{\boldsymbol{v}_l}[h(H(\pi_\theta), \gamma)] - \mathbb{E}_{\boldsymbol{v}_l}[h(H(\pi_\theta), u(\gamma))]\ | \\ \leq & \max(\rho, 1-\rho) \sup_{\gamma \in \mathcal{H}} ||\gamma - h(\gamma)|| \leq \frac{\delta}{3}. \\ & |\hat{J}_l(\gamma) - \hat{J}_l(u(\gamma))| \\ = & \frac{1}{n_l}\ |\sum_{i=1}^{n_l} \Big(h(H(\pi_{\theta_i}), \gamma) - h(H(\pi_{\theta_i}), u(\gamma)) \Big)\ | \\ \leq & \max(\rho, 1-\rho) \sup_{\gamma \in \mathcal{H}} ||\gamma - u(\gamma)|| \leq \frac{\delta}{3}. \end{aligned}$$

As $H(\cdot) \subseteq [H_{min}, H_{max}]$, we can bound the probability that $|J_l(u(\gamma)) - \hat{J}_l(u(\gamma))| > \delta/3$ for any $\delta \geq 0$ by Hoeffding's bound:

$$Pr\big(|J_l(u(\gamma)) - \hat{J}_l(u(\gamma))| \geq \frac{\delta}{3}\big) \leq 2e^{-\frac{2n_l\delta^2}{9|H_{max} - H_{min}|^2}}.$$

As card$(\mathcal{U}) = k < \infty$, we can bound the probability that $|J_l(h_i) - \hat{J}_l(h_i)| < \frac{\delta}{3}$ holds for all $h_i \in \mathcal{U}$ with the union bound:

$$\begin{aligned} & Pr\big(\max_{h_i \in \mathcal{U}} |J_l(h_i) - \hat{J}_l(h_i)| \geq \frac{\delta}{3}\big) \\ \leq & \sum_{i=1}^k Pr\big(|J_l(h_i) - \hat{J}_l(h_i)| \geq \frac{\delta}{3}\big) \leq 2ke^{-\frac{2n_l\delta^2}{9|H_{max} - H_{min}|^2}}. \end{aligned}$$

Therefore with probability at least $(1 - 2ke^{-\frac{2n_l\delta^2}{9|H_{max} - H_{min}|^2}})$,

$$|J_l(\gamma) - \hat{J}_l(\gamma)| \leq \frac{\delta}{3} + \frac{\delta}{3} + \frac{\delta}{3} = \delta$$

holds uniformly for all $\gamma \in \mathcal{H}$. Therefore

$$\begin{aligned} & \sum_{l=1}^{\infty} Pr(\sup_{\gamma \in \mathcal{H}} |J_l(\gamma) - \hat{J}_l(\gamma)| > \delta) \\ \leq & \sum_{l=1}^{\infty} 2ke^{-\frac{2n_l\delta^2}{9|H_{max} - H_{min}|^2})} < \infty. \end{aligned}$$

The last inequality holds as $n_l = \Theta(l^\beta)$ and $\beta > 0$ by Assumption (2b). By Borel-Cantelli Lemma, $Pr(\sup_{\gamma \in \mathcal{H}} |J_l(\gamma) - \hat{J}_l(\gamma)| > \delta\ i.o.) = 0$. As the above proof holds for any $\delta > 0$, $\sup_{\gamma \in \mathcal{H}} |J_l(\gamma) - \hat{J}_l(\gamma)| \to 0$ as $l \to \infty$ with probability 1. In other words, $\hat{J}_l(\cdot)$ converges uniformly to $J_l(\cdot)$ as $l \to \infty$ with probability 1. Note that this uniform convergence holds whenever Assumption (2b) and (2e) hold.

Then we prove that $\lim_{l \to +\infty} |\hat{\xi}_l - \xi(\rho, \boldsymbol{v}_l)| = 0$, with probability 1.

Since $\sup_{\gamma \in \mathcal{H}} |J_l(\gamma) - \hat{J}_l(\gamma)| \to 0$ as $l \to \infty$ with probability 1, for any $\varepsilon > 0$, there exists some $L(\varepsilon) > 0$ such that $\sup_{\gamma \in \mathcal{H}} |J_l(\gamma) - \hat{J}_l(\gamma)| < \varepsilon$ holds for all $l > L(\varepsilon)$, with probability 1. Therefore with probability 1 and $l > L(\varepsilon)$,

$$J_l(\hat{\xi}_l) - \varepsilon < \hat{J}_l(\hat{\xi}_l), \quad \hat{J}_l(\xi(\rho, \boldsymbol{v}_l)) < J_l(\xi(\rho, \boldsymbol{v}_l)) + \varepsilon.$$

As $\xi(\rho, \boldsymbol{v}_l)$ minimizes $J_l(\cdot)$ and $\hat{\xi}_l$ minimizes $\hat{J}_l(\cdot)$, we have

$$J_l(\xi(\rho, \boldsymbol{v}_l)) \leq J_l(\hat{\xi}_l), \quad \hat{J}_l(\hat{\xi}_l) \leq \hat{J}_l(\xi(\rho, \boldsymbol{v}_l)).$$

Combining the above two equalities, we get

$$\begin{aligned} J_l(\xi(\rho, \boldsymbol{v}_l)) - \varepsilon \leq J_l(\hat{\xi}_l) - \varepsilon < \hat{J}_l(\hat{\xi}_l) \\ \leq \hat{J}_l(\xi(\rho, \boldsymbol{v}_l)) < J_l(\xi(\rho, \boldsymbol{v}_l)) + \varepsilon. \end{aligned}$$

Therefore for any $\varepsilon > 0$ and $l > L(\varepsilon)$,

$$J_l(\xi(\rho, \boldsymbol{v}_l)) - \varepsilon < J_l(\hat{\xi}_l) < J_l(\xi(\rho, \boldsymbol{v}_l)) + \varepsilon$$

with probability 1. Equivalently, $J_l(\hat{\xi}_l) - J_l(\xi(\rho, \boldsymbol{v}_l)) \to 0$ as $l \to +\infty$ with probability 1.

We define $J_{\boldsymbol{v}}$ in the same way as we defined J_l, namely,

$$J_{\boldsymbol{v}}(\gamma) := \mathbb{E}_{\boldsymbol{v}}[h(H(\pi_\theta), \gamma)]$$

for all $\boldsymbol{v} \in \mathcal{V}$ and $\gamma \in \mathcal{H}$. By Assumption (2c), the ρ-quantile of $\{H(\pi_\theta) : \theta \sim f_{\boldsymbol{v}}(\cdot)\}$ is unique for all $\boldsymbol{v} \in \mathcal{V}$, i.e., $J_{\boldsymbol{v}}(\gamma)$ is minimized with a unique $\xi(\rho, \boldsymbol{v})$ for all $\boldsymbol{v} \in \mathcal{V}$. We can verify from the definition of $J_{\boldsymbol{v}}(\cdot)$ such that if $\gamma \leq \xi(\rho, \boldsymbol{v})$,

$$\begin{aligned} J_{\boldsymbol{v}}(\gamma) - J_{\boldsymbol{v}}(\xi(\rho, \boldsymbol{v})) = \mathbb{E}_{\boldsymbol{v}}\big[(H(\pi_\theta) - \gamma)\mathbf{1}_{[\gamma, \xi(\rho, \boldsymbol{v}))}(H(\pi_\theta))\big] \\ + (\xi(\rho, \boldsymbol{v}) - \gamma)\big(Pr_{\boldsymbol{v}}(H(\pi_\theta) \geq \xi(\rho, \boldsymbol{v})) - (1-\rho)\big). \end{aligned}$$

If $\gamma > \xi(\rho, \boldsymbol{v})$,

$$\begin{aligned} J_{\boldsymbol{v}}(\gamma) - J_{\boldsymbol{v}}(\xi(\rho, \boldsymbol{v})) = \mathbb{E}_{\boldsymbol{v}}\big[(\gamma - H(\pi_\theta))\mathbf{1}_{(\xi(\rho, \boldsymbol{v}), \gamma)}(H(\pi_\theta))\big] \\ + (\gamma - \xi(\rho, \boldsymbol{v}))(Pr_{\boldsymbol{v}}(H(\pi_\theta) \leq \xi(\rho, \boldsymbol{v})) - \rho). \end{aligned}$$

By definition of $\xi(\rho, \boldsymbol{v})$, it holds that

$$\begin{aligned} Pr_{\boldsymbol{v}}\big(H(\pi_\theta) \geq \xi(\rho, \boldsymbol{v})\big) - (1-\rho) \geq 0, \\ Pr_{\boldsymbol{v}}\big(H(\pi_\theta) \leq \xi(\rho, \boldsymbol{v})\big) - \rho \geq 0. \end{aligned}$$

Therefore for any $\boldsymbol{v} \in \mathcal{V}$, $J_{\boldsymbol{v}}(\gamma)$ decreases monotonically if $\gamma < \xi(\rho, \boldsymbol{v})$ and increases monotonically if $\gamma > \xi(\rho, \boldsymbol{v})$. Since the global minimizer is always unique, $J_{\boldsymbol{v}}(\gamma) > J_{\boldsymbol{v}}(\xi(\rho, \boldsymbol{v}))$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2020.3015931, IEEE Transactions on Automatic Control

for any $\gamma \neq \xi(\rho, \boldsymbol{v})$. For any fixed $\delta' > 0$ and any $\boldsymbol{v} \in \mathcal{V}$, $\rho \in (0, 1)$, if $|\gamma - \xi(\rho, \boldsymbol{v})| \geq \delta'$, it holds that

$$\begin{aligned} &|J_{\boldsymbol{v}}(\gamma) - J_{\boldsymbol{v}}(\xi(\rho, \boldsymbol{v}))| \\ \geq& \min\big(J_{\boldsymbol{v}}(\xi(\rho, \boldsymbol{v}) + \delta') - J_{\boldsymbol{v}}(\xi(\rho, \boldsymbol{v})), \\ &J_{\boldsymbol{v}}(\xi(\rho, \boldsymbol{v}) - \delta') - J_{\boldsymbol{v}}(\xi(\rho, \boldsymbol{v}))\big). \end{aligned}$$

For any fixed $\delta' > 0$ and all $\boldsymbol{v} \in \mathcal{V}$, it holds that

$$\begin{aligned} &J_{\boldsymbol{v}}(\xi(\rho, \boldsymbol{v}) + \delta') - J_{\boldsymbol{v}}(\xi(\rho, \boldsymbol{v})) > 0, \text{ and} \\ &J_{\boldsymbol{v}}(\xi(\rho, \boldsymbol{v}) - \delta') - J_{\boldsymbol{v}}(\xi(\rho, \boldsymbol{v})) > 0. \end{aligned}$$

Since $\mathcal{V}$ is compact, we get

$$\begin{aligned} &\inf_{\boldsymbol{v}\in\mathcal{V}} \big(J_{\boldsymbol{v}}(\xi(\rho, \boldsymbol{v}) + \delta') - J_{\boldsymbol{v}}(\xi(\rho, \boldsymbol{v}))\big) > 0, \text{ and} \\ &\inf_{\boldsymbol{v}\in\mathcal{V}} \big(J_{\boldsymbol{v}}(\xi(\rho, \boldsymbol{v}) - \delta') - J_{\boldsymbol{v}}(\xi(\rho, \boldsymbol{v}))\big) > 0 \end{aligned}$$

for any $\delta' > 0$.

Assume that $\hat{\xi}_l - \xi(\rho, \boldsymbol{v}_l)$ does not converge to 0 with probability 1. Then there exists $\bar{\delta} > 0$ such that $Pr(\{|\hat{\xi}_l - \xi(\rho, \boldsymbol{v}_l)| \geq \bar{\delta} \ i.o.\}) > 0$. Since $J_l(\hat{\xi}_l) - J_l(\xi(\rho, \boldsymbol{v}_l)) \to 0$, we know that with positive probability, there exists a subsequence $\{l_k\}_{k\geq 0} \in \mathbb{N}^\infty$ such that $|\hat{\xi}_{l_k} - \xi(\rho, \boldsymbol{v}_{l_k})| \geq \bar{\delta}$ for each $k \in \mathbb{N}$ and $\lim_{k\to\infty}(J_{l_k}(\hat{\xi}_{l_k}) - J_{l_k}(\xi(\rho, \boldsymbol{v}_{l_k}))) = 0$. However,

$$\begin{aligned} &|J_{l_{k_j}}(\hat{\xi}_{l_k}) - J_{l_k}(\xi(\rho, \boldsymbol{v}_{l_k}))| \\ \geq& \min\Big(\inf_{\boldsymbol{v}\in\mathcal{V}} \big(J(\xi(\rho, \boldsymbol{v}) - \bar{\delta}) - J(\xi(\rho, \boldsymbol{v}))\big), \\ &\inf_{\boldsymbol{v}\in\mathcal{V}} \big(J(\xi(\rho, \boldsymbol{v}) + \bar{\delta}) - J(\xi(\rho, \boldsymbol{v}))\big)\Big) > 0, \end{aligned}$$

which contradicts our assumption that $\lim_{k\to\infty}(J_{l_k}(\hat{\xi}_{l_k}) - J_{l_k}(\xi(\rho, \boldsymbol{v}_{l_k}))) = 0$. Therefore the assumption is wrong and $\lim_{l\to+\infty} |\hat{\xi}_l - \xi(\rho, \boldsymbol{v}_l)| = 0$ with probability 1. □

We can now give a proof to Lemma 3.

Proof. By Assumption (2e), $\inf_{\pi\in\Pi} G(\pi) > 0$. By definition of $(1-\rho)$-quantile, it holds for any $\boldsymbol{v} \in \mathcal{V}$ that

$$\mathbb{E}_{\boldsymbol{v}}[G(\pi_\theta)S(\pi_\theta, \boldsymbol{v}, \rho)] \geq \inf_{\pi\in\Pi} G(\pi)\rho > 0.$$

Similarly we can show

$$\sum_{i=1}^{n_l} G(\pi_{\theta_i})\hat{S}(\pi_{\theta_i}, \boldsymbol{v}, \rho) \geq \inf_{\pi\in\Pi} G(\pi) > 0.$$

There are two types of approximation involved in b_l: the first is to approximate $\xi_H(\rho, \boldsymbol{v}_l)$ and $\xi_U(1-\rho, \boldsymbol{v}_l)$ by $\hat{\xi}_{H,l}$ and $\hat{\xi}_{U,l}$. The second is to approximate the expectations with sample means, for example, to approximate $\mathbb{E}_{\boldsymbol{v}_l}[G(\pi_\theta)\hat{S}(\pi_{\theta_i}, \boldsymbol{v}_l, \rho)\Gamma(\theta)]$ with $\frac{1}{n_l}\sum_{i=1}^{n_l} G(\pi_{\theta_i})\hat{S}(\pi_{\theta_i}, \boldsymbol{v}_l, \rho)\Gamma(\theta_i)$.

We have shown that $\lim_{l\to\infty} |\xi_H(\rho, \boldsymbol{v}_l) - \hat{\xi}_{H,l}| = 0$ with probability 1 and $\lim_{l\to\infty} |\xi_U(1-\rho, \boldsymbol{v}_l) - \hat{\xi}_{U,l}| = 0$ with probability 1 by Lemma 4. With the continuous approximation of δ as explained in Remark 3, we can show $\lim_{l\to\infty} |S(\pi_\theta, \boldsymbol{v}_l, \rho) - \hat{S}(\pi_\theta, \boldsymbol{v}_l, \rho)| = 0$ with probability 1. using the continuous mapping theorem. We only need to consider the second part in this proof.

$\Gamma(\cdot)$ is bounded as it is a continuous function defined over a compact set (by Assumption (2d)). By Assumption (2e), both G and H are bounded over Π. Therefore $\lim_{l\to\infty}\Big|\frac{1}{n_l}\sum_{i=1}^{n_l} G(\pi_{\theta_i})\hat{S}(\pi_{\theta_i}, \boldsymbol{v}_l, \rho)\Gamma(\theta_i) - \frac{1}{n_l}\sum_{i=1}^{n_l} G(\pi_{\theta_i})S(\pi_{\theta_i}, \boldsymbol{v}_l, \rho)\Gamma(\theta_i)\Big| = 0$ with probability 1.

As $G(\pi_\theta)$, $S(\pi_\theta, \boldsymbol{v}_l, \rho)$, $\Gamma(\theta)$ are all bounded for any θ and ρ, there exist finite a, b such that $a \leq G(\pi_\theta)S(\pi_\theta, \boldsymbol{v}_l, \rho)\Gamma(\theta) \leq b$ for any $\theta \in \Theta$. By Hoeffding's inequality, for any $\varepsilon > 0$

$$\begin{aligned} &Pr\Big(\Big|\frac{1}{n_l}\sum_{i=1}^{n_l} G(\pi_{\theta_i})S(\pi_{\theta_i}, \boldsymbol{v}_l, \rho)\Gamma(\theta_i) \\ &\quad - \mathbb{E}_{\boldsymbol{v}_l}[G(\pi_\theta)S(\pi_\theta, \boldsymbol{v}_l, \rho)\Gamma(\theta)]\Big| \geq \varepsilon\Big) \leq 2e^{\frac{-2n_l\varepsilon^2}{(b-a)^2}}. \end{aligned}$$

By Assumption (2b), $n_l = \Theta(l^\beta)$ and $\beta > 0$. Therefore for any $\varepsilon > 0$,

$$\begin{aligned} &\sum_{l=1}^{\infty} Pr\Big(\Big|\frac{1}{n_l}\sum_{i=1}^{n_l} G(\pi_{\theta_i})S(\pi_{\theta_i}, \boldsymbol{v}_l, \rho)\Gamma(\theta_i) \\ &- \mathbb{E}_{\boldsymbol{v}_l}[G(\pi_\theta)S(\pi_\theta, \boldsymbol{v}_l, \rho)\Gamma(\theta)]\Big| \geq \varepsilon\Big) \leq \sum_{l=1}^{\infty} 2e^{\frac{-2n_l\varepsilon^2}{(b-a)^2}} < \infty. \end{aligned}$$

Then by Borel-Cantelli Lemma, with probability 1,

$$\begin{aligned} \lim_{l\to\infty} \Big|&\frac{1}{n_l}\sum_{i=1}^{n_l} G(\pi_{\theta_i})S(\pi_{\theta_i}, \boldsymbol{v}_l, \rho)\Gamma(\theta_i) \\ &- \mathbb{E}_{\boldsymbol{v}_l}[G(\pi_\theta)S(\pi_\theta, \boldsymbol{v}_l, \rho)\Gamma(\theta)]\Big| = 0. \end{aligned}$$

Similarly, we can show that with probability 1,

$$\begin{aligned} \lim_{l\to\infty} \Big|&\frac{1}{n_l}\sum_{i=1}^{n_l} G(\pi_{\theta_i})S(\pi_{\theta_i}, \boldsymbol{v}_l, \rho) \\ &- \mathbb{E}_{\boldsymbol{v}_l}[G(\pi_\theta)S(\pi_\theta, \boldsymbol{v}_l, \rho)]\Big| = 0. \end{aligned}$$

Then $\lim_{l\to\infty} b_l = 0$ holds with probability 1 by continuous mapping theorem.

□

Now we provide a proof for Theorem 1.

Proof. We connect the sequence $\{\hat{\eta}_l\}_{l\geq 0}$ to the ODE (12) by applying Theorem 2. We need to verify that all sufficient conditions in 2 hold properly. By (15), $\hat{\eta}_{l+1} - \hat{\eta}_l = \alpha_l\Big(\tilde{L}(\boldsymbol{v}_l; \rho) + b_l + w_l\Big)$.

- By Assumption (2a), $\tilde{L}(\boldsymbol{v}; \rho)$ is continuous in $\boldsymbol{v} \in int(\mathcal{V})$. Since $m^{-1}(\eta)$ is continuous in η, $\tilde{L}(\boldsymbol{v}; \rho)\Big|_{\boldsymbol{v}=m^{-1}(\eta)}$ is continuous in η. (12) has a unique integral curve by Assumption (2a).
- By Assumption (2b), $\{\alpha_l\}_{l\geq 0}$ is bounded and decreasing.
- By Lemma 3, $\lim_{l\to\infty} b_l = 0$ with probability 1 with Assumption (2b), (2c), (2d), (2e).
- Then we show that for any $N \in \mathbb{N}^+$, $\lim_{l\to\infty}\Big(\sup_{k:\sum_{i=n}^{k}\alpha_i<N} ||\sum_{i=n}^{k}\alpha_i w_i||\Big) = 0$. Define $M_l = \sum_{i=1}^{l}\alpha_i w_i$. Then $M_l = M_{l-1} + \alpha_n w_n$. As the set $\{\theta_i\}_{i=1}^{n_l}$ is generated i.i.d. with distribution $f_{m^{-1}(\hat{\eta}_l)}(\cdot)$ and $\hat{\eta}_l = \mathbb{E}_{m^{-1}(\hat{\eta}_l)}[\Gamma(\theta)]$, it holds that

$$\begin{aligned} &\mathbb{E}[M_l|M_1, \ldots, M_{l-1}] - M_{l-1} \\ =&(1-\alpha_l)\lambda_l\Big(\mathbb{E}_{m^{-1}(\hat{\eta}_l)}[\frac{1}{n_l}\sum_{i=1}^{n_l}\Gamma(\theta_i)|M_{l-1}] - \hat{\eta}_l\Big) = 0 \end{aligned}$$

regardless of the value of $\hat{\eta}_l$. To show that $\{M_n\}_{n\geq 0}$ is a martingale, we show that $\mathbb{E}[||M_n||] < \infty$. Note that w_i is independent on w_j if $i \neq j$, as all θ are independently generated. Therefore $\mathbb{E}[w_i^\intercal w_j] = \mathbb{E}[w_i]^\intercal \mathbb{E}[w_j] = 0$.

$$\begin{aligned}&\mathbb{E}[||M_n||^2]\\ =&\mathbb{E}[M_n^\intercal M_n] = \mathbb{E}[(\sum_{i=1}^n \alpha_i w_i)^\intercal(\sum_{i=1}^n \alpha_i w_i)]\\ =&\sum_{i=1}^n \alpha_i^2 \mathbb{E}[w_i^\intercal w_i] + \sum_{i=1}^n \sum_{j\neq i} \alpha_i \alpha_j \mathbb{E}[w_i^\intercal w_j]\\ =&\sum_{i=1}^n \alpha_i^2 \mathbb{E}[w_i^\intercal w_i] = \sum_{i=1}^n \frac{(1-\alpha_i)^2 \lambda_i^2}{n_i} \mathrm{Cov}_{m^{-1}(\hat{\eta}_i)}[\Gamma(\theta)].\end{aligned}$$

As $\Gamma(\theta)$ is continuous and the domain Θ is compact, there exists $0 < C < \infty$ such that $\mathrm{Cov}_{\boldsymbol{v}}[\Gamma(\theta)] \leq C$ for any $\boldsymbol{v} \in \mathcal{V}$. Therefore by Assumption (2b),

$$\mathbb{E}[||M_n||^2] \leq \sum_{i=1}^n C \frac{(1-\alpha_i)^2 \lambda_i^2}{n_i} = O(\sum_{l=1}^n \frac{1}{l^{\beta+2\lambda}}).$$

By Assumption (2b), $\beta + 2\lambda > 1$. Therefore $\lim_{n\to\infty} \mathbb{E}[||M_n||^2] < \infty$. As $\{||M_n||^2\}$ increases monotonically, we know $\sup_n \mathbb{E}[||M_n||^2] = \lim_{n\to\infty} \mathbb{E}[||M_n||^2] < \infty$. Since and $\mathbb{E}[||M_n||] \leq \sqrt{\mathbb{E}[||M_n||^2]}$, it holds that $\sup_n \mathbb{E}[||M_n||] < \infty$ and $\{M_n\}_{n\geq 0}$ is a martingale. Then by L_2 martingale convergence theorem, there exists M_∞ such that $M_n \to M_\infty$ with probability 1 and $\mathbb{E}[||M_\infty||^2] < \infty$.

$$\begin{aligned}&\sup_{\{k:\sum_{i=n}^k \alpha_i < N\}} ||\sum_{i=n}^k \alpha_i w_i||\\ =&\sup_{\{k:\sum_{i=n}^k \alpha_i < N\}} ||M_k - M_{n-1}|| \leq 2 \sup_{k\geq n} ||M_k||.\end{aligned}$$

Therefore

$$\begin{aligned}0 &\leq \lim_{n\to\infty} \Big(\sup_{\{k:\sum_{i=n}^k \alpha_i < N\}} ||\sum_{i=n}^k \alpha_i w_i|| \Big)\\ &\leq \lim_{n\to\infty} \Big(2 \sup_{k\geq n-1} ||M_k|| \Big) = 0\end{aligned}$$

for any finite $N > 0$.

Since all conditions in Theorem 2 are satisfied, the limit set of sequence $\{\hat{\eta}_l\}_{l\geq 0}$ is a internally chain recurrent connected set for the flow induced by $\tilde{L}(m^{-1}(\eta); \rho)$ with probability 1. □

To further interpret Theorem 1, we first note that any equilibrium of (12) forms an internally chain recurrent set by itself. The following result shows a sufficient condition for an equilibrium point $\bar{\boldsymbol{v}}^*$ of (12) to be locally asymptotically stable, which means that there exists a small neighborhood of $\bar{\boldsymbol{v}}^*$ such that once entered, (13) will converge to $\bar{\boldsymbol{v}}^*$.

Theorem 3. *Let $\varphi : \mathcal{V} \to \mathbb{R}$ be any function such that $\frac{\partial}{\partial \boldsymbol{v}} \varphi(\boldsymbol{v}) = \tilde{L}(\boldsymbol{v}; \rho)$. Any equilibrium $\bar{\boldsymbol{v}}^* \in int(\mathcal{V})$ of* (13) *that is an isolated local maximum of $\varphi(\boldsymbol{v})$ is locally asympototically stable.*

Proof. The Lyapunov function we use is similar to that in [47]:

$$V(\boldsymbol{v}) :=\varphi(\bar{\boldsymbol{v}}^*) - \varphi(\boldsymbol{v}),$$

where $\bar{\boldsymbol{v}}^*$ is an isolated local maximum of $\varphi(\boldsymbol{v})$ and $\boldsymbol{v}$ is in some neighborhood of $\bar{\boldsymbol{v}}^*$ such that $\varphi(\bar{\boldsymbol{v}}^*) \geq \varphi(\boldsymbol{v})$ and $V(\boldsymbol{v}) \geq 0$. By previous analysis, $\log \varphi(\boldsymbol{v})$ and $V(\boldsymbol{v})$ are continuous in $\boldsymbol{v}$. For the derivative:

$$\frac{dV(\boldsymbol{v})}{dt} = -\frac{\partial \boldsymbol{v}}{\partial t} \frac{\partial \varphi(\boldsymbol{v})}{\partial \boldsymbol{v}} = -\Big(\tilde{L}(\boldsymbol{v}; \rho)\Big)^\intercal (\mathrm{Cov}[\Gamma(\theta)])^{-1} \tilde{L}(\boldsymbol{v}; \rho).$$

As $\mathrm{Cov}_{\boldsymbol{v}}[\Gamma(\theta)]$ is positive definite for $\boldsymbol{v} \in int(\mathcal{V})$, $\big(\mathrm{Cov}_{\boldsymbol{v}}[\Gamma(\theta)]\big)^{-1}$ is also positive definite. Therefore $\frac{\partial V(\boldsymbol{v})}{\partial t} \leq 0$ in a neighborhood of $\boldsymbol{v}^*$ and $\frac{\partial V(\boldsymbol{v})}{\partial t} = \mathbf{0}$ if and only if $\tilde{L}(\boldsymbol{v}; \rho) = \mathbf{0}$, which guarantees that $\boldsymbol{v}$ is a stationary point of (13). As $\bar{\boldsymbol{v}}^*$ is an isolated local maximum of $\varphi(\boldsymbol{v})$, it is the only stationary point in some neighborhood of $\bar{\boldsymbol{v}}^*$. Therefore $\frac{\partial V(\boldsymbol{v})}{\partial \boldsymbol{v}} = \mathbf{0}$ if and only if $\boldsymbol{v} = \bar{\boldsymbol{v}}^*$ (if $\boldsymbol{v}$ is in the neighborhood of $\boldsymbol{v}^*$) and $\bar{\boldsymbol{v}}^*$ is locally asymptotically stable. □

The proof of Theorem 3 shows that $\varphi(\boldsymbol{v})$ always decreases in the interior of $\mathcal{V}$ unless it hits a stationary point of (13), which suggests a stronger property of our algorithm as stated in Theorem 4. In order to state the result we need to first introduce some definitions. By Assumption (2a), $Z := \Big(\tilde{L}(\boldsymbol{v}; \rho)\Big)^\intercal (\mathrm{Cov}_{\boldsymbol{v}}[\Gamma(\theta)])^{-1}$ is a continuous vector field defined on $\mathcal{V} \subset \mathbb{R}^{d_v}$ with unique integral curves. The *flow* of Z is the family of mappings $\{\Phi_t(\cdot)\}_{t\in\mathbb{R}}$ defined on $\mathcal{V}$ by $\frac{\partial \Phi_t(\boldsymbol{v})}{\partial t} = Z(\Phi_t(\boldsymbol{v}))$ such that $\Phi_0(\boldsymbol{v}) \equiv \boldsymbol{v}$ and $\Phi_t(\Phi_s(\boldsymbol{v})) \equiv \Phi_{t+s}(\boldsymbol{v})$ for any $\boldsymbol{v} \in \mathcal{V}$, $t, s \in \mathbb{R}$. $\boldsymbol{v} \in \mathcal{V}$ is an *equilibrium* if $\Phi_t(\boldsymbol{v}) = \boldsymbol{v}$ for all t. A set $\mathcal{V}' \subset \mathcal{V}$ is *positively invariant* under the flow Φ if for all $t \geq 0$, $\Phi_t(\mathcal{V}') = \mathcal{V}'$.

Theorem 4. *If all equilibria of* (13) *are isolated, the sequence $\{\boldsymbol{v}_l\}_{l\geq 0}$ derived by Algorithm 1 converges toward an equilibrium of* (13) *as $l \to \infty$ with probability 1.*

Proof. Let φ be defined in the same way as in Theorem 3. We first show that φ is bounded over $\mathcal{V}$. By definition of $\tilde{L}(\boldsymbol{v}; \rho)$ in (9), $\tilde{L}(\boldsymbol{v}; \rho) = \frac{\mathbb{E}_{\boldsymbol{v}}[G(\pi_\theta) S(\pi_\theta, \boldsymbol{v}, \rho) \Gamma(\theta)]}{L(\boldsymbol{v}; \rho)} - m(\boldsymbol{v})$. Since G has a positive lower bound (by Assumption (2e)) and $\mathbb{E}_{\boldsymbol{v}}[S(\pi_\theta, \boldsymbol{v}', \rho)] \geq \rho$ for any $\boldsymbol{v} \in \mathcal{V}$, $L(\boldsymbol{v}; \rho) \geq \inf_{\pi\in\Pi} G(\pi)\rho > 0$. Since Γ is continuous over Θ, Θ and $\mathcal{V}$ are compact (by Assumption (2d)), $\Gamma(\theta)$ and $m(\boldsymbol{v}) = \mathbb{E}_{\boldsymbol{v}}[\Gamma(\theta)]$ are both bounded. Since G is also bounded (by Assumption (2e)), $\mathbb{E}_{\boldsymbol{v}}[G(\pi_\theta) S(\pi_\theta, \boldsymbol{v}, \rho) \Gamma(\theta)]$ is also bounded over $\mathcal{V}$ for any $\rho \in (0, 1)$. Therefore φ is also bounded over $\mathcal{V}$.

Let Φ be a flow induced by (13) and Λ be the set of all equilibria of (13). By definition, Λ is positively invariant under Φ. Define $V : \mathcal{V} \to \mathbb{R}^{\geq 0}$ as $V(\boldsymbol{v}) := \sup_{\boldsymbol{v}'\in\mathcal{V}} \varphi(\boldsymbol{v}') - \varphi(\boldsymbol{v})$. $\sup_{\boldsymbol{v}'\in\mathcal{V}} \varphi(\boldsymbol{v}') < \infty$ as φ is shown to be bounded in $\mathcal{V}$. By definition of Λ and the proof of Theorem 3, the mapping $t \mapsto V(\Phi_t(\boldsymbol{v}))$ is constant-valued for $\boldsymbol{v} \in \Lambda$ and strictly decreasing for $\boldsymbol{v} \in int(\mathcal{V})\backslash\Lambda$. Since we also assume that (13) has only isolated equilibria and $\boldsymbol{v}$ is always in the interior of $\mathcal{V}$ (Assumption (2f)), $\{\boldsymbol{v}_l\}_{l\geq 0}$ converges to an equilibrium of (13) as $l \to \infty$ with probability 1 by Corollary 3.3 in [46]. □

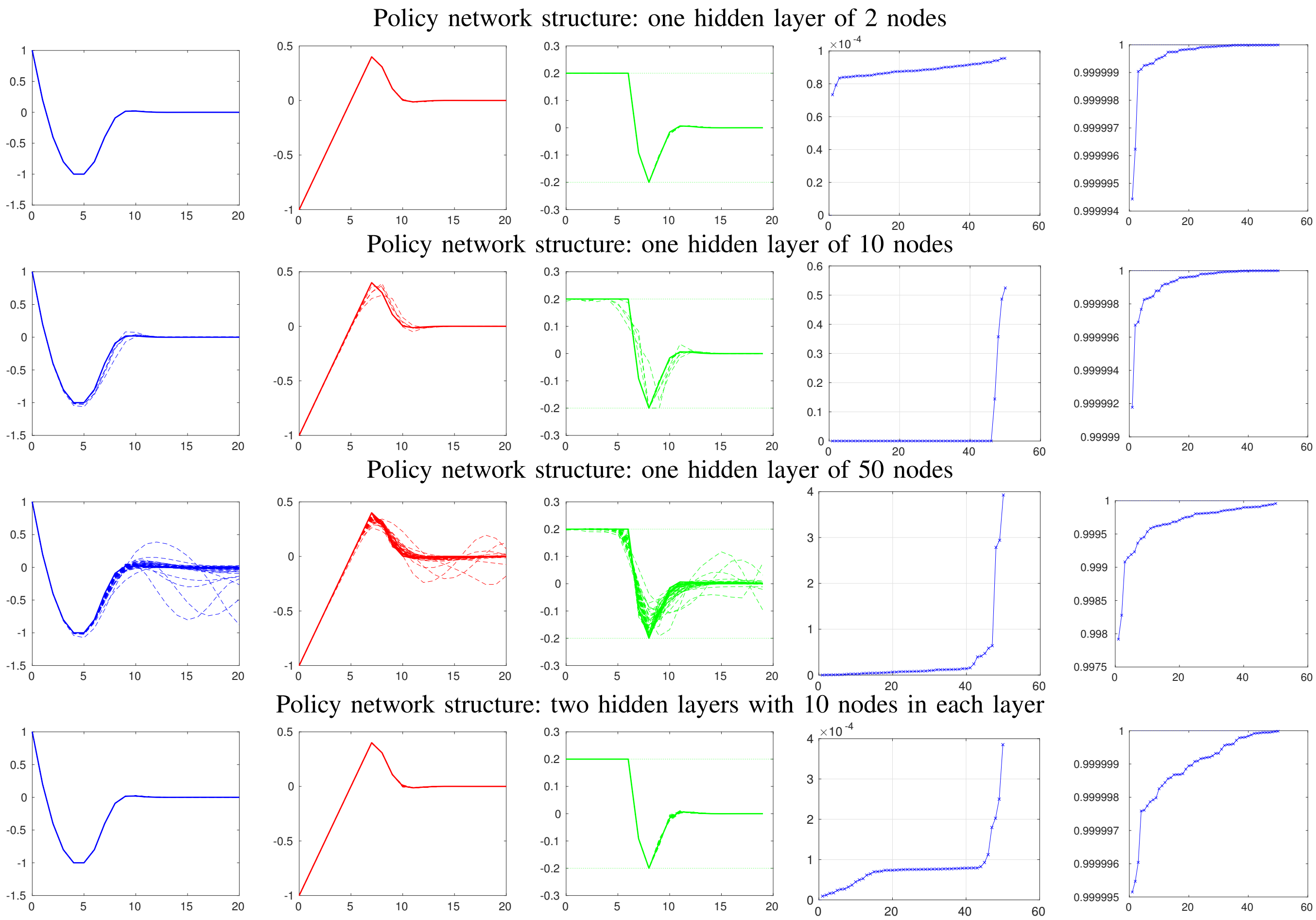


Fig. 1: Comparison of the globally optimal policy π^* and the 50 learned policies for different policy network structures. Each row corresponds to a policy network structure. From left to right, the first three columns represent the trajectories of the two states $\boldsymbol{x}_t(1)$, $\boldsymbol{x}_t(2)$ and the input $\boldsymbol{u}_t$ over time t. The solid line in each figure is for π^* and the dashed lines are for the learned policies. In the fourth column, we show the gap between their G-values and $G(\pi^*)$ in ascending order. In the last column, we show the H-values of the learned policies in ascending order.

V. Experimental Results

We show the performance of CCE in two numerical experiments: One is a discrete-time finite-horizon constrained linear quadratic regulator problem and the other is a 2D robot navigation problem with only local observations.

A. Constrained Linear Quadratic Regulator

We first run CCE on a simple finite-horizon constrained linear quadratic regulator (LQR) problem. The problem is convex and thus can be solved efficiently and accurately. The goal of this example is to check if CCE can converge to the globally optimal solution for a convex problem, as well as the effect of policy network structure on the performance of CCE.

Given an initial state $\boldsymbol{x}_0 \in \mathbb{R}^{n_{\boldsymbol{x}}}$, a finite horizon $N \in \mathbb{N}^+$, a lower bound $\boldsymbol{u}_{low} \in \mathbb{R}^{n_{\boldsymbol{u}}}$ and an upper bound $u_{upp} \in \mathbb{R}^{n_{\boldsymbol{u}}}$ of inputs, the optimization problem to be solved is

$$\begin{aligned} \min_{\substack{\boldsymbol{u}_0,\dots,\boldsymbol{u}_{N-1}\\ \boldsymbol{x}_1,\dots,\boldsymbol{x}_N}} \quad & \sum_{t=0}^{N-1} \left(\boldsymbol{x}_{t+1}^\intercal Q \boldsymbol{x}_{t+1} + \boldsymbol{u}_t^\intercal R \boldsymbol{u}_t\right) \\ s.t. \quad & \boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + B\boldsymbol{u}_t,\ \forall t = 0,\dots,N-1, \\ & \boldsymbol{u}_{low} \preccurlyeq \boldsymbol{u}_t \preccurlyeq \boldsymbol{u}_{upp},\ \forall t = 0,\dots,N-1, \\ & \boldsymbol{x}_t \in \mathbb{R}^{n_{\boldsymbol{x}}},\ \boldsymbol{u}_t \in \mathbb{R}^{n_{\boldsymbol{u}}},\ \forall t = 0,\dots,N-1. \end{aligned} \tag{17}$$

It is well known that if $Q \succeq 0$ and $R \succ 0$, an optimal solution $\boldsymbol{u}_t^*$ to (17) at each time $t = 0,\dots,N-1$ is a continuous piecewise affine function of the state $\boldsymbol{x}_t$ [48]. At each time t, there exists a polyhedral partition $\{P_t^j\}$, $j = 1,\dots,k_t$ of $\mathbb{R}^{n_{\boldsymbol{x}}}$ such that $P_t^j = \{\boldsymbol{x} \in \mathbb{R}^{n_{\boldsymbol{x}}} | F_t^j \boldsymbol{x} \le K_t^j\}$ and $\boldsymbol{u}_t^*(\boldsymbol{x}) = C_t^j \boldsymbol{x} + d_t^j$ for $\boldsymbol{x} \in P_t^j$. Since Problem (17) is convex, we can compute its globally optimal solution $\boldsymbol{x}_t^*$ and $\boldsymbol{u}_t^*$ via tools such as CVX [49].

The specific matrices we used are

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, R = \begin{bmatrix} 0.3 \end{bmatrix},$$
$$\boldsymbol{u}_{low} = -0.2, \boldsymbol{u}_{upp} = 0.2, \boldsymbol{x}_0 = \begin{bmatrix} 1 & -1 \end{bmatrix}^\intercal.$$

TABLE I: $J_i(\tau)$, $Z_i(\tau)$ and constraint upper bound d_i for $i = 1, 2, 3, 4$, $\tau \in (S \times A)^N$.

i	$J_i(\tau)$	$Z_i(\tau)$	d_i	J_i Markovian	Z_i Markovian
1	1 for each state in $\mathcal{G}$; $2\lvert y\rvert$ for each state with $y \in [-2, -0.2]$; 0 otherwise.	-1 if the robot arrives $\mathcal{G}$ which is absorbing; 0 otherwise.	-0.5	Yes	Yes
2	30 times the minimum signed distance from any state in τ to $\mathcal{B}$.	-1 if the robot visited $\mathcal{G}$ in τ; 0 otherwise.	-0.5	No	No
3	Same as $J_2(\tau)$.	-1 for each state in $\mathcal{G}$; 0 otherwise.	-5	No	Yes
4	Same as $J_1(\tau)$.	-1 if the robot visits $\mathcal{G}$ and never visits $\mathcal{B}$; 0 otherwise.	-0.5	Yes	No

The horizon length is $N = 20$, which is long enough to for $\boldsymbol{u}^*$ to drive the states back to the origin. The state and input trajectories derived by a globally optimal policy π^* are shown in Figure 1.

We now solve (17) using CCE. We define J as the objective function in (17) and the constraint function Z as follows:

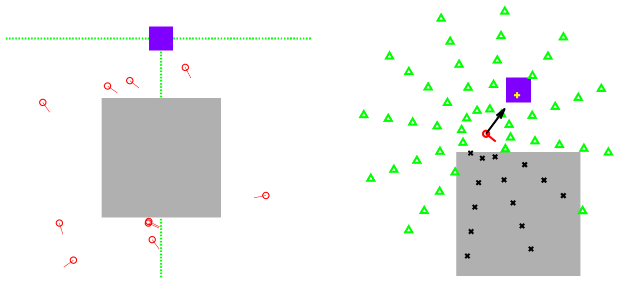
$$\begin{aligned} & Z(\boldsymbol{x}_0, \boldsymbol{u}_0, \ldots, \boldsymbol{x}_{N-1}, \boldsymbol{u}_{N-1}, \boldsymbol{x}_N) \\ =& 1 - \max_t \max(\boldsymbol{u}_{low} - \boldsymbol{u}_t, \boldsymbol{u}_t - \boldsymbol{u}_{upp}, 0). \end{aligned}$$

Therefore, a trajectory $\tau = \boldsymbol{x}_0, \boldsymbol{u}_0, \ldots, \boldsymbol{x}_{N-1}, \boldsymbol{u}_{N-1}, \boldsymbol{x}_N$ is feasible if and only if $Z(\tau) \geq 1$.

We use a fully-connected neural network to represent the policy or controller, which maps from the current position $\boldsymbol{x}_t = [\boldsymbol{x}_t(1), \boldsymbol{x}_t(2)]^\intercal \in \mathbb{R}^2$ to an input $\boldsymbol{u}_t \in \mathbb{R}$. We compare four different policy network structures: three networks with one hidden layer of 2, 10 or 50 nodes respectively and one network with two hidden layers of 10 nodes in each layer. The activation function for each hidden layer is the rectified linear unit (ReLU) and thus the learned controller is also a piecewise linear function of the states. There is no activation function for the output layer. Note that the policy represented by the neural network is time-invariant and thus it may be impossible to reach the globally optimal objective value.

We assume that $F_{\mathcal{V}}$ is a family of Gaussian distributions with diagonal covariance matrices. Each sample policy is represented as a vector composed of all its network weights. For each policy network, we repeatedly run CCE for 50 times. At the beginning of each experiment, we randomly initialize the policy distribution parameter $\boldsymbol{v} \in \mathcal{V}$. In each iteration, we draw 100 sample policies from the current policy distribution. The results are shown in Figure 1, which includes the state and input trajectories of both the globally optimal policy π^* and each learned policy $\hat{\pi}$, the suboptimality gap $G(\hat{\pi}) - G(\pi^*)$ of the G-value and the H-value for each learned policy.

CCE converged in all experiments. The performance of the learned policy is largely dependent on the architecture of the policy network. As this example problem is simple, it turns out that a neural network with a single hidden layer of 2 nodes can approximate the globally optimal policy accurately and consistently. As we increase the number of nodes in the hidden layer, it becomes more difficult to find or converge to feasible solutions; the suboptimality gap of G-value also increases. Meanwhile, neither the number of hidden nodes nor the number of weight parameters is a reliable metric to evaluate the complexity of the model. As the policy network has 2 inputs and 1 output, a network with a single layer of 50 nodes has 201 weight parameters and a network with two hidden layers of 10 nodes has 151 weight parameters. However, Figure 1 shows that the performance of the latter network is much better than the previous one: all the trajectories led by the 50 learned policies are very close to that generated by π^* and the suboptimality gap of G-value is very small.



(a) (b)

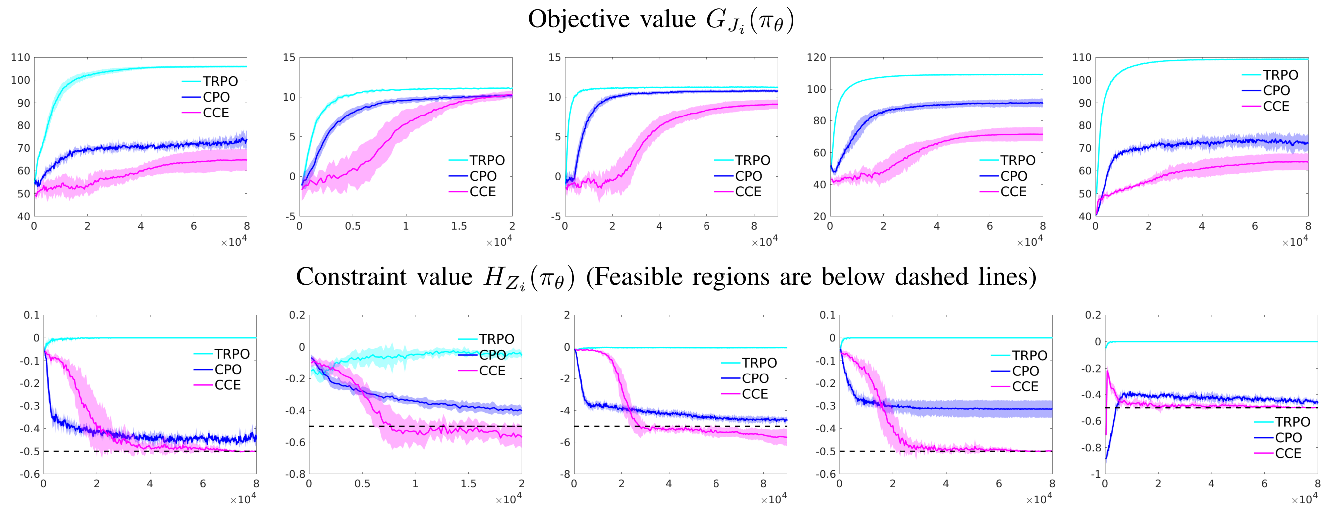
Fig. 2: (2a) Map of the 2D navigation example. There are one obstacle region (grey rectangle), one goal region (blue rectangle) and 10 randomly selected initial states (red circles pointing to the forward direction). Dotted lines are added to show x and y axes. (2b) Illustration of the local features in the robot's local coordinate at one of the initial states, with $n_s = 5$. Obstacle nodes, goal nodes and free nodes are labeled by black crosses, yellow plus signs and green triangles respectively. The goal direction (black arrow) is also included in local features.

B. 2D Navigation

We also consider a mobile robot navigation task with only local observations. Unlike the previous example for which we can reliably compute the globally optimal solution, the optimization problem in this example is non-convex and we have to resort to approximate solutions. The goal is to compare the performance of CCE with that of the constrained policy optimization algorithm, which is a state-of-the-art constrained RL algorithm [34].

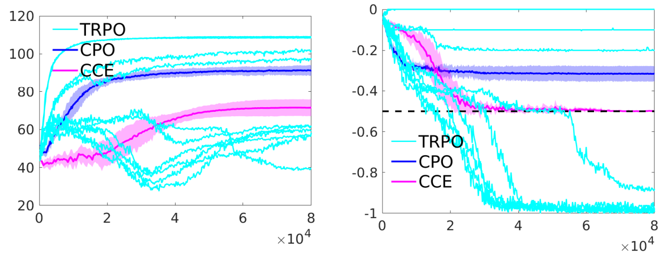
The robot's state space is $S = \{(x, y, \zeta) \big| x_{min} \leq x \leq x_{max}, y_{min} \leq y \leq y_{max}, -\pi \leq \zeta < \pi\}$, which contains the robot's position and orientation in the global coordinate. The action space is 2-dimensional: $A = \{(v, \omega) \big| |v| \leq v_{max}, |\omega| \leq \omega_{max}\}$, which are linear and angular speed respectively. The environment map is shown in Figure 2a, where there is a compact goal region $\mathcal{G}$ and a disjoint compact obstacle region

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2020.3015931, IEEE Transactions on Automatic Control



(a) $i = 1$ (b) $i = 2$ (c) $i = 3$ (d) $i = 4$ (e) $i = 1$

Fig. 3: Learning curves of CCE, CPO and TRPO with different objectives G_{J_i} and constraints H_{Z_i}. The horizontal axes show the total number of sample trajectories for CCE and the total number of equivalent sample trajectories for TRPO and CPO. The vertical axes show the sample mean of the objective and constraint values of the learned policy (for TRPO and CPO) or the learned policy distribution (for CCE). The shade shows 1 standard deviation. The region below the dashed line in the second row is feasible. Each experiment is repeated for 5 times.



(a) $G_{J_4}(\pi_\theta)$. (b) $H_{Z_4}(\pi_\theta)$.

Fig. 4: Average performance of CCE, CPO and TRPO for Experiment 4 with initial feasible policy.

$\mathcal{B}$. The overall goal of the navigation task is to reach the goal region $\mathcal{G}$ without colliding with the obstacle region $\mathcal{B}$, while the objective and constraint are encoded in four different ways as shown in Table I.

The policy is again modeled as a fully connected neural network with 2 hidden layers and 30 nodes in each layer. The activation function is ReLU for hidden layers and the hyperbolic tangent function (tanh) for the output layer. The policy network maps from local observations to actions. The local observations are interpreted as follows.

We assume that the robot cannot observe (x, y, ζ) directly and can only use local sensors (shown in Figure 2b) to observe if $\mathcal{B}$ or $\mathcal{G}$ is in its neighborhood and the direction of the center of $\mathcal{G}$ in its local coordinate. For a given parameter $n_s \in \mathbb{N}^+$ and sampling time Δt, we design a radial grid as n_s circles in the robot's local coordinate. The difference between the diameters of adjacent circles is $v_{max}\Delta t > 0$. There are $\lceil 2\pi/\omega_{max} \rceil$ uniformly distributed observation points on each circle and the robot can measure the label for each node. An observation point is labeled: 1, if it belongs to $\mathcal{G}$; -1, if it belongs to $\mathcal{B}$; and 0, otherwise. We also assume that the robot can sense the direction of the center of $\mathcal{G}$ in its local coordinate without knowing the distance. In total, there are a total of $(2 + n_s\lceil 2\pi/\omega_{max} \rceil)$ outputs of the local observation model. In our experiment, $\omega_{max} = \frac{\pi}{6}$, $n_s = 5$, so there are 62 local observations as the inputs to the policy network.

We compare the performance of CCE to trust region policy optimization (TRPO) [4], a state-of-the-art unconstrained RL algorithm, and its variant for constrained problems called constrained policy optimization (CPO). The policy space Π_Θ is a set of deterministic stationary policies. Trajectory length for all experiments is set to $N = 30$. Each sampled policy is evaluated using 10 sample trajectories. We set $\rho = 0.2$. For TRPO and CPO, we set the batch size as 6,000, the discount factor as 0.999, and the step size for trust region as 0.01. All other parameters are the default values in rllab [50].

We show the learning curves of CCE, TRPO and CPO for each experiment in Figure 3. For experiments in which J_i is not strictly positive, we use $\exp(J_i)$ instead of J_i to update the policy distributions in CCE. The vertical axes in Figure 3 show the *average* objective and constraint values of the learned policy. For CCE, the average values are computed with all rollout trajectories that are simulated with *all* the policies sampled at the current iteration. For CPO and TRPO, we simulate the current policy from exactly the same set of initial states and compute the average objective and constraint values for all trajectories.

Results by TRPO show that the constraints cannot be satisfied by merely optimizing the corresponding objectives. However, CCE successfully outputs feasible policies in all experiments. On the other hand, CPO needs significantly more samples to find a single feasible policy, or simply converges to an infeasible policy especially if the constraint is non-Markovian.

One may argue that CPO is designed to work with feasible initial policies and Markovian objectives and constraints (specifically, both J and Z are discounted total rewards). Thus, we repeat the first experiment ($i = 1$) with feasible initial policies and obtain the result in the last column of Figure 3. In this case, CPO leaves the feasible region rapidly and then follows generally the same path as if it is initialized with an infeasible policy. This behavior suggests that its incapability to enforce constraint satisfaction is not due to the lack of initial feasibility. Although CCE also leaves the feasible region at an early stage of iterations, it regains feasibility much faster than the previous case with infeasible initial polices. These results suggest that CCE is more reliable than CPO for applications where the strict constraint satisfaction is critical.

In Figure 4, we compare the performance of CPO and CCE in Experiment 4 to that of TRPO with objective $G_{J_4} - 100H_{Z_4}$. The fixed penalty coefficient 100 is chosen to be neither too large nor too small so it can show a large variety of locally optimal behaviors with very different G_{J_4}-values and H_{Z_4}-values. Figure 4 clearly shows the trade-off between G_{J_4}-values and H_{Z_4}-values, which partially explains the gap between the G_{J_4}-value outputs of CCE and CPO. With a fixed penalty coefficient, the policies learned by TRPO are either infeasible or with very small constraint values. The policy output by CCE has higher G_{J_4}-value than all the feasible policies found by TRPO and CPO.

VI. Conclusions and Discussions

In this work, we studied a safe RL problem with constraints defined as the expected cost over finite-length trajectories. We proposed a constrained cross-entropy-based method to solve this problem, analyzed its asymptotic performance using an ODE and proved its convergence. We showed with simulation experiments that our method converges to the global optimum in a convex problem and can effectively learn feasible policies without assumptions on the feasibility of initial policies with both Markovian and non-Markovian objective functions and constraint functions.

CCE also has several limitations. First, CCE is expected to be less sample-efficient than gradient-based methods especially for problems with high-dimensional action spaces. Unlike gradient-based methods such as TRPO, CCE does not infer the performance of unseen policies from previous experience. As a result, it has to repetitively sample good policies in order to make steady improvement. Meanwhile, CCE can be easily parallelized as each sampled policy is evaluated independently, which may effectively mitigate the problem of high sample complexity as other evolutionary methods [51]. Second, CCE requires accurate evaluation of the G-values and H-values of all sample policies. If the variance of G-value and H-value estimation is comparable with the range of G-values or H-values, the ranking of the estimated G-values or H-values of the sample policies cannot reliably imply the ranking of their true values. As a result, there will be an enduring gap between the average performance of the elite sample policies and that of all sample polices, which prevents CCE from converging. Third, CCE cannot be directly generalized to problems with multiple constraints. In order to deal with more than one constraint, we need to either define a ranking function over different constraints, or manually define a function to map all the constraint values into a single scalar.

Given all these limitations, we find the CCE method to be particularly useful in learning hierarchical policies. With a high-level policy that specifies intermediate goals and thus reduces the state space for low-level policies, we can use CCE to train a (locally) optimal low-level policy while satisfying local constraints. As shown in the experiment of our paper, CCE converges with reasonable sample complexity and outperforms CPO on its constraint performance. Since the satisfaction of low-level constraints is of critical significance to the performance of the overall policy, CCE seems to be especially well-suited for this application.

References

[1] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol. II*, 3rd ed. Athena Scientific, 2007.

[2] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” in *Reinforcement Learning*. Springer, 1992, pp. 5–32.

[3] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, “Deterministic policy gradient algorithms,” in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 387–395.

[4] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, “Trust region policy optimization,” in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 1889–1897.

[5] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, “High-dimensional continuous control using generalized advantage estimation,” *arXiv preprint arXiv:1506.02438*, 2015.

[6] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” in *International Conference on Machine Learning*, 2016, pp. 1928–1937.

[7] S. Gu, T. Lillicrap, I. Sutskever, and S. Levine, “Continuous deep q-learning with model-based acceleration,” in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 2829–2838. [Online]. Available: http://proceedings.mlr.press/v48/gu16.html

[8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.

[9] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” *arXiv preprint arXiv:1312.5602*, 2013.

[10] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[11] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, “Mastering the game of go with deep neural networks and tree search,” *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[12] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, “Mastering the game of go without human knowledge,” *Nature*, vol. 550, no. 7676, p. 354, 2017.

[13] I. Zamora, N. G. Lopez, V. M. Vilches, and A. H. Cordero, “Extending the openai gym for robotics: a toolkit for reinforcement learning using ros and gazebo,” *arXiv preprint arXiv:1608.05742*, 2016.

[14] I. Popov, N. Heess, T. Lillicrap, R. Hafner, G. Barth-Maron, M. Vecerik, T. Lampe, Y. Tassa, T. Erez, and M. Riedmiller, “Data-efficient deep reinforcement learning for dexterous manipulation,” *arXiv preprint arXiv:1704.03073*, 2017.

[15] S. Levine, C. Finn, T. Darrell, and P. Abbeel, “End-to-end training of deep visuomotor policies,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[16] T. Zhang, G. Kahn, S. Levine, and P. Abbeel, “Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search,” in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 528–535.

[17] W. Montgomery, A. Ajay, C. Finn, P. Abbeel, and S. Levine, “Reset-free guided policy search: Efficient deep reinforcement learning with stochastic initial states,” in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 3373–3380.

[18] J. Hu, M. C. Fu, S. I. Marcus *et al.*, “A model reference adaptive search method for stochastic global optimization,” *Communications in Information and Systems*, vol. 8, no. 3, pp. 245–276, 2008.

[19] J. Garcıa and F. Fernández, “A comprehensive survey on safe reinforcement learning,” *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437–1480, 2015.

[20] T. M. Moldovan and P. Abbeel, “Safe exploration in markov decision processes,” in *Proceedings of the 29th International Coference on International Conference on Machine Learning*. Omnipress, 2012, pp. 1451–1458.

[21] M. Turchetta, F. Berkenkamp, and A. Krause, “Safe exploration in finite markov decision processes with gaussian processes,” in *Advances in Neural Information Processing Systems*, 2016, pp. 4312–4320.

[22] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, “Safe reinforcement learning via shielding,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[23] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, “Learning-based model predictive control for safe exploration,” in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 6059–6066.

[24] K. P. Wabersich and M. N. Zeilinger, “Safe exploration of nonlinear dynamical systems: A predictive safety filter for reinforcement learning,” *arXiv preprint arXiv:1812.05506*, 2018.

[25] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, “Safe model-based reinforcement learning with stability guarantees,” in *Advances in neural information processing systems*, 2017, pp. 908–918.

[26] M. Pirotta, M. Restelli, A. Pecorino, and D. Calandriello, “Safe policy iteration,” in *International Conference on Machine Learning*, 2013, pp. 307–315.

[27] E. Uchibe and K. Doya, “Constrained reinforcement learning from intrinsic and extrinsic rewards,” in *Development and Learning, 2007. ICDL 2007. IEEE 6th International Conference on*. IEEE, 2007, pp. 163–168.

[28] Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone, “Risk-constrained reinforcement learning with percentile risk criteria,” *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6070–6120, 2017.

[29] M. Yu, Z. Yang, M. Kolar, and Z. Wang, “Convergent policy optimization for safe reinforcement learning,” in *Advances in Neural Information Processing Systems*, 2019, pp. 3121–3133.

[30] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh, “A lyapunov-based approach to safe reinforcement learning,” in *Advances in neural information processing systems*, 2018, pp. 8092–8101.

[31] Y. Chow, O. Nachum, A. Faust, E. Duenez-Guzman, and M. Ghavamzadeh, “Lyapunov-based safe policy optimization for continuous control,” *arXiv preprint arXiv:1901.10031*, 2019.

[32] E. Altman, “Asymptotic properties of constrained markov decision processes,” *Mathematical Methods of Operations Research*, vol. 37, no. 2, pp. 151–170, 1993.

[33] G. Dalal, K. Dvijotham, M. Vecerik, T. Hester, C. Paduraru, and Y. Tassa, “Safe exploration in continuous action spaces,” *arXiv preprint arXiv:1801.08757*, 2018.

[34] J. Achiam, D. Held, A. Tamar, and P. Abbeel, “Constrained policy optimization,” in *International Conference on Machine Learning*, 2017, pp. 22–31.

[35] S. Mannor, R. Rubinstein, and Y. Gat, “The cross entropy method for fast policy search,” in *In International Conference on Machine Learning*. Morgan Kaufmann, 2003, pp. 512–519.

[36] I. Szita and A. Lörincz, “Learning tetris using the noisy cross-entropy method,” *Learning*, vol. 18, no. 12, 2006.

[37] M. Kobilarov, “Cross-entropy randomized motion planning,” in *Robotics: Science and Systems*, 2011.

[38] S. C. Livingston, E. M. Wolff, and R. M. Murray, “Cross-entropy temporal logic motion planning,” in *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*. ACM, 2015, pp. 269–278.

[39] I. Papusha, J. Fu, U. Topcu, and R. M. Murray, “Automata theory meets approximate dynamic programming: Optimal control with temporal logic constraints,” in *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE, 2016, pp. 434–440.

[40] J. Fu, I. Papusha, and U. Topcu, “Sampling-based approximate optimal control under temporal logic constraints,” in *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*. ACM, 2017, pp. 227–235.

[41] L. Li and J. Fu, “Sampling-based approximate optimal temporal logic planning,” in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1328–1335.

[42] T. Homem-de Mello, “A study on the cross-entropy method for rare-event probability estimation,” *INFORMS Journal on Computing*, vol. 19, no. 3, pp. 381–394, 2007.

[43] J. Hu, P. Hu, and H. S. Chang, “A stochastic approximation framework for a class of randomized optimization algorithms,” *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 165–178, 2012.

[44] R. Y. Rubinstein and B. Melamed, *Modern simulation and modeling*. Wiley New York, 1998, vol. 7.

[45] S. Kullback and R. A. Leibler, “On information and sufficiency,” *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[46] M. Benaim, “A dynamical system approach to stochastic approximations,” *SIAM Journal on Control and Optimization*, vol. 34, no. 2, pp. 437–472, 1996.

[47] A. G. Joseph and S. Bhatnagar, “Revisiting the cross entropy method with applications in stochastic global optimization and reinforcement learning,” in *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands*, 2016, pp. 1026–1034.

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

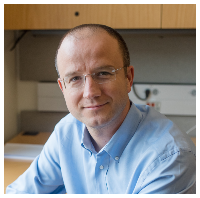
[49] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 2.1,” http://cvxr.com/cvx, Mar. 2014.

[50] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, “Benchmarking deep reinforcement learning for continuous control,” in *International Conference on Machine Learning*, 2016, pp. 1329–1338.

[51] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, “Evolution strategies as a scalable alternative to reinforcement learning,” *arXiv preprint arXiv:1703.03864*, 2017.

Min Wen received her Ph.D. degree from the University of Pennsylvania in 2019 and the B.Eng degree from Zhejiang University, China in 2013. She currently works at Google LLC, with the goal of solving large-scale practical problems with learning-based techniques.

Her research interests include reinforcement learning, control theory and formal methods, with a special focus on providing high-level task implementation guarantees for learning-based controllers.



Ufuk Topcu joined the Department of Aerospace Engineering at the University of Texas at Austin as an assistant professor in Fall 2015. He received his Ph.D. degree from the University of California at Berkeley in 2008. He held research positions at the University of Pennsylvania and California Institute of Technology.

His research focuses on the theoretical, algorithmic and computational aspects of design and verification of autonomous systems through novel connections between formal methods, learning theory and controls.