CONSERVATIVE SAFETY CRITICS FOR EXPLORATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Safe exploration presents a major challenge in reinforcement learning (RL): when active data collection requires deploying partially trained policies, we must ensure that these policies avoid catastrophically unsafe regions, while still enabling trial and error learning. In this paper, we target the problem of safe exploration in RL by learning a conservative safety estimate of environment states through a critic, and provably upper bound the likelihood of catastrophic failures at every training iteration. We theoretically characterize the tradeoff between safety and policy improvement, show that the safety constraints are satisfied with high probability during training, derive provable convergence guarantees for our approach which is no worse asymptotically than standard RL, and demonstrate the efficacy of the proposed approach on a suite of challenging navigation, manipulation, and locomotion tasks. Our results show that the proposed approach can achieve competitive task performance while incurring significantly lower catastrophic failure rates during training than prior methods. Videos are at this url https://sites.google.com/view/safe-exploration/

1 INTRODUCTION

Reinforcement learning (RL) is a powerful framework for learning-based control because it can enable agents to learn to make decisions automatically through trial and error. However, in the real world, the cost of those trials – and those errors – can be quite high: an aerial robot that attempts to fly at high speed might initially crash, and then be unable to attempt further trials due to extensive physical damage. However, learning complex skills without any failures at all is likely impossible. Even humans and animals experience regularly experience failure, but quickly learn from their mistakes and behave *cautiously* in risky situations. In this paper, our goal is to develop safe exploration methods for RL that similarly exhibit *conservative* behavior, erring on the side of caution in particularly dangerous settings, and minimizing the number of catastrophic failures.

A number of previous approaches have tackled this problem of safe exploration, often by formulating the problem as a constrained Markov decision process (CMDP) (Garcıa & Fernández, 2015; Altman, 1999). However most of these approaches require additional assumptions, like assuming access to a function that can be queried to check if a state is safe (Thananjeyan et al., 2020), assuming access to a default safe controller (Koller et al., 2018; Berkenkamp et al., 2017), assuming knowledge of all the unsafe states (Fisac et al., 2019), and only obtaining safe policies at optimality, while being unsafe during the training process (Tessler et al., 2018; Dalal et al., 2018).

In this paper, we propose a general safe RL algorithm, with safety guarantees *throughout* training. Our method only assumes access to a sparse (e.g., binary) indicator for *catastrophic failure*, in the standard RL setting. Our method trains a *conservative* safety critic that overestimates the probability of catastrophic failure, building on tools in the recently proposed conservative Q-learning framework (Kumar et al., 2020) for offline RL. In order to bound the likelihood of catastrophic failures at every iteration, we impose a KL-divergence constraint on successive policy updates so that the stationary distribution of states induced by the old and the new policies are not arbitrarily different. Based on the safety critic's value, we consider a chance constraint denoting probability of failure, and optimize the policy through primal-dual gradient descent.

Our key contributions in this paper are designing an algorithm that we refer to as *Conservative Safety Critics* (CSC), that learns a conservative estimate of how safe a state is, using this conservative estimate for safe-exploration and policy updates, and theoretically providing safety guarantees

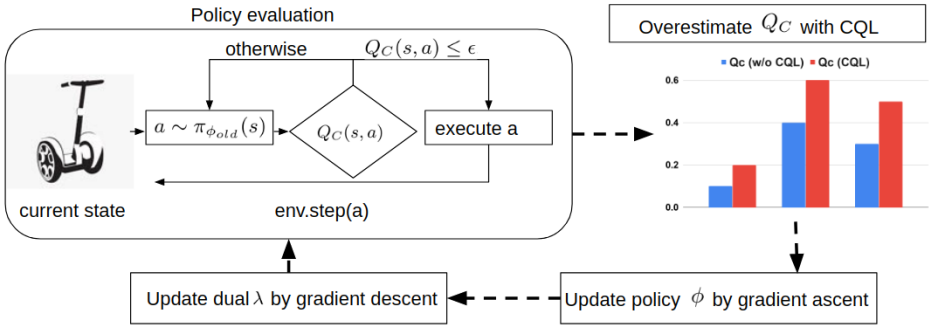


Figure 1: Illustrative figure of the approach described in Algorithm 1.

throughout training. Through empirical evaluation in five separate simulated robotic control domains, we show that CSC is able to learn effective policies while reducing the rate of catastrophic failures by up to 50% over prior safe exploration methods.

2 Preliminaries

We describe the problem setting of a constrained MDP (Altman, 1999) specific to our approach and the conservative Q learning (Kumar et al., 2020) framework that we build on in our algorithm.

Constrained MDPs. A constrained MDP (CMDP) is a tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma, \mu, \mathcal{C})$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is a transition kernel, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a task reward function, $\gamma \in (0, 1)$ is a discount factor, μ is a starting state distribution, and $\mathcal{C} = \{(c_i : \mathcal{S} \to \{0, 1\}, \chi_i \in \mathbb{R}) | i \in \mathbb{Z}\}$ is a set of (safety) constraints that the agent must satisfy, with constraint functions c_i taking values either 0 (*alive*) or 1 (*failure*) and limits χ_i defining the maximal allowable amount of non-satisfaction, in terms of expected probability of failure. A stochastic policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ is a mapping from states to action distributions, and the set of all stationary policies is denoted by Π. Let Π_C denote the set of feasible policies that satisfy the constraints. The reinforcement learning problem in a CMDP is to find the optimal policy within Π_C that maximizes the discounted sum of returns. Without loss of generality, we can consider a *single* constraint, where $\mathcal{C}$ denotes the constraint satisfaction function $C : \mathcal{S} \to \{0, 1\}$, similar to the task reward function, and a limit χ. We define the discounted future state distribution of a policy π as $d^\pi(s) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi)$, the state value function as $V_R^\pi(s) = \mathbb{E}_{\tau\sim\pi} [R(\tau)|s_0 = s]$, the state-action value function as $Q_R^\pi(s, a) = \mathbb{E}_{\tau\sim\pi} [R(\tau)|s_0 = s, a_0 = a]$, and the advantage function as $A_R^\pi(s, a) = Q_R^\pi(s, a) - V_R^\pi(s)$. we define similar quantities with respect to the constraint function, as V_C, Q_C, and A_C. So, we have $V_R^\pi(\mu) = \mathbb{E}_{\tau\sim\pi} [\sum_{t=0}^{\infty} R(s_t, a_t)]$ and $V_C^\pi(\mu) = \mathbb{E}_{\tau\sim\pi} [\sum_{t=0}^{\infty} C(s_t, a_t)]$. When the policy is parameterized as π_ϕ, we will denote $d^\pi(s)$ as $\rho_\phi(s)$.

Conservative Q Learning. CQL (Kumar et al., 2020) is a method for offline/batch RL (Lange et al., 2012; Levine et al., 2020) that aims to learn a Q-function such that the expected value of a policy under the learned Q function lower bounds its true value, preventing over-estimation due to out-of-distribution actions as a result. In addition to training Q-functions via standard Bellman error, CQL minimizes the expected Q-values under a particular distribution of actions, $\mu(a|s)$, and maximizes the expected Q-value under the on-policy distribution, $\pi(a|s)$. CQL in and off itself might lead to unsafe exploration, whereas we will show in Section 3, how the theoretical tool introduced in CQL can be used to devise a safe RL algorithm.

3 The Conservative Safe-Exploration Framework

In this section we describe our safe exploration framework. The safety constraint $C(s)$ defined in Section 2 is an indicator of catastrophic failure: $C(s) = 1$ when a state s is unsafe and $C(s) = 0$ when it is not, and we ideally desire $C(s) = 0 \; \forall s \in \mathcal{S}$ that the agent visits. Since we do not make any assumptions in the problem structure for RL, we cannot guarantee this, but can at best reduce the *probability of failure* in every episode. So, we formulate the constraint as $V_C^\pi(\mu) = \mathbb{E}_{\tau\sim\pi} [\sum_{t=0}^{\infty} C(s_t, a_t)] \leq \chi$, where $\chi \in [0, 1)$ denotes *probability of failure*. Our approach is

motivated by the insight that by being "conservative" with respect to how safe a state is, and hence by over-estimating this probability of failure, we can effectively ensure constrained exploration.

Figure 1 provides an overview of the approach. The key idea of our algorithm is to train a conservative safety critic denoted as Q_C, that overestimates how unsafe a particular state is and modifies the exploration strategy to appropriately account for this safety under-estimate (by overestimating the probability of failure). During policy evaluation in the environment, we use the safety critic $Q_C(s, a)$ to reduce the chance of catastrophic failures by checking whether taking action a in state s has $Q_C(s, a)$ less than a threshold ϵ. If not, we re-sample a from the current policy $\pi(a|s)$.

We now discuss our algorithm more formally. We start by discussing the procedure for learning the safety critic Q_C, then discuss how we incorporate this in the policy gradient updates, and finally discuss how we perform safe exploration during policy execution in the environment.

Overall objective. Our objective is to learn an optimal policy π^* that maximizes task rewards, while respecting the constraint on expected probability of failures.

$$\pi^* = \arg\max_{\pi \in \Pi_C} \; V_R^{\pi}(\mu) \quad \text{where} \quad \Pi_C = \{\pi \in \Pi : V_C^{\pi}(\mu) \leq \chi\} \tag{1}$$

Learning the safety critic. The safety critic Q_C is used to obtain an estimate of how unsafe a particular state is, by providing an estimate of *probability of failure*, that will be used to guide exploration. We desire the estimates to be "conservative", in the sense that the probability of failure should be an *over-estimate* of the actual probability so that the agent can err on the side of caution while exploring. To train such a critic Q_C, we incorporate tools from CQL to estimate Q_C through updates similar to those obtained by reversing the sign of α in Equation 2 of CQL($\mathcal{H}$) (Kumar et al., 2020). This gives us an *upper bound* on Q_C instead of a lower bound, as guaranteed by CQL. Formally the safety critic is trained via the following objective, where the objective inside $\arg\min$ is called $CQL(\zeta)$, ζ parameterizes Q_C, and k denotes the kth update iteration.

$$\begin{aligned} \hat{Q}_C^{k+1} \leftarrow \arg\min_{Q_C} \alpha \left(-\mathbb{E}_{s\sim\mathcal{D}_{env}, a\sim\pi_\phi(a|s)}[Q_C(s,a)] + \mathbb{E}_{(s,a)\sim\mathcal{D}_{env}}[Q_C(s,a)]\right) \\ + \frac{1}{2}\mathbb{E}_{(s,a,s',c)\sim\mathcal{D}_{env}}\left[\left(Q_C(s,a) - \hat{\mathcal{B}}^{\pi_\phi}\hat{Q}_C^k(s,a)\right)^2\right] \end{aligned} \tag{2}$$

For states sampled from the replay buffer $\mathcal{D}_{env}$, the first term seeks to maximize the expectation of Q_C over actions sampled from the current policy, while the second term seeks to minimize the expectation of Q_C over actions sampled from the replay buffer. $\mathcal{D}_{env}$ can include off-policy data, and also offline-data (if available). In our experiments in Section ??, we demonstrate how $\mathcal{D}_{env}$ can be seeded with offline data to bootstrap the learning of Q_C. We interleave the gradient descent updates for training of Q_C, with gradient ascent updates for policy π_ϕ and gradient descent updates for Lagrange multiplier λ, which we describe next.

Policy learning. Since we want to learn policies that obey the constraint we set in terms of the safety critic, we solve the objective in equation 1 via a surrogate policy improvement problem as described below:

$$\begin{aligned} &\max_{\pi_\phi} \mathbb{E}_{s\sim\rho_{\phi_{old}}, a\sim\pi_\phi}\left[A_R^{\pi_{\phi_{old}}}(s,a)\right] \\ &s.t. \quad \mathbb{E}_{s\sim\rho_{\phi_{old}}}[D_{\text{KL}}(\pi_{\phi_{old}}(\cdot|s)||\pi_\phi(\cdot|s))] \leq \delta \quad \text{and} \quad V_C^{\pi_\phi}(\mu) \leq \chi \end{aligned} \tag{3}$$

Here, we have introduced a D_{KL} constraint to ensure successive policies are *close* in order to help obtain guaranteed safety bounds in Section 4. We replace the $D_{\text{KL}}(\pi_{\phi_{old}}(\cdot|s)||\pi_\phi(\cdot|s))$ term by its second order Taylor expansion (expressed in terms of the Fisher Information Matrix) and enforce the resulting constraint exactly (Schulman et al., 2015). For enforcing the constraint on $V_C^{\pi_\phi}(\mu)$, we following the primal-dual optimization method of Lagrange multipliers without making any simplifications of the constraint term $V_C^{\pi_\phi}(\mu)$. This, as per equation (9) can be rewritten as

$$\begin{aligned} &\max_{\pi_\phi} \mathbb{E}_{s\sim\rho_{\phi_{old}}, a\sim\pi_\phi}\left[A_R^{\pi_{\phi_{old}}}(s,a)\right] \quad s.t. \quad V_C^{\pi_{\phi_{old}}}(\mu) + \frac{1}{1-\gamma}\mathbb{E}_{s\sim\rho_{\phi_{old}}, a\sim\pi_\phi}\left[A_C(s,a)\right] \leq \chi \\ &s.t. \quad \mathbb{E}_{s\sim\rho_{\phi_{old}}}[D_{\text{KL}}(\pi_{\phi_{old}}(\cdot|s)||\pi_\phi(\cdot|s))] \leq \delta \end{aligned} \tag{4}$$

Algorithm 1 CSC: safe exploration with conservative safety critics

Initialize V_θ^r (task value fn), Q_ζ^s (safety critic), policy π_ϕ, λ, $\mathcal{D}_{env}$, thresholds ϵ, δ, χ.
Set $\hat{V}_C^{\pi_{\phi_{old}}}(\mu) \leftarrow \chi$. ▷ $\hat{V}_C^{\pi_{\phi_{old}}}(\mu)$ denotes avg. failures in the *previous* epoch.
for epochs until convergence **do** ▷ Execute actions in the environment. Collect on-policy samples.
  for episode e in $\{1, \ldots, \text{M}\}$ **do**
    Set $\epsilon \leftarrow (1-\gamma)(\chi - \hat{V}_C^{\pi_{\phi_{old}}}(\mu))$
    Sample $a \sim \pi_{\phi_{old}}(s)$. Execute a iff $Q_C(s,a) \leq \epsilon$. Else, resample a.
    Obtain next state s', $r = R(s,a)$, $c = C(s')$.
    $\mathcal{D}_{env} \leftarrow \mathcal{D}_{env} \cup \{(s, a, s', r, c)\}$ ▷ If available, $\mathcal{D}_{env}$ can be seeded with off-policy/offline data
  end for
  Store the average episodic failures $\hat{V}_C^{\pi_{\phi_{old}}}(\mu) \leftarrow \sum_{e=1}^{M} \hat{V}_C^e$
  for step t in $\{1, \ldots, \text{N}\}$ **do** ▷ Policy and Q function updates using $\mathcal{D}_{env}$
    Gradient ascent on ϕ and (Optionally) add Entropy regularization (equation 7)
    Gradient updates for the Q-function $\zeta := \zeta - \eta_Q \nabla_\zeta CQL(\zeta)$
    Gradient descent step on Lagrange multiplier λ (equation 9)
  end for
  $\phi_{old} \leftarrow \phi$
end for

We consider the Lagrangian dual of this constrained problem, which we can solve *approximately* by alternating gradient descent as shown below.

$$\max_{\pi_\phi} \min_{\lambda \geq 0} \mathbb{E}_{s\sim\rho_{\phi_{old}}, a\sim\pi_\phi}\left[A_R^{\pi_{\phi_{old}}}(s,a)\right] - \lambda\left(V_C^{\pi_{\phi_{old}}}(\mu) + \frac{1}{1-\gamma}\mathbb{E}_{s\sim\rho_{\phi_{old}}, a\sim\pi_\phi}\left[A_C(s,a)\right] - \chi\right)$$
$$s.t. \quad \frac{1}{2}(\phi - \phi_{old})^T \boldsymbol{F}(\phi - \phi_{old}) \leq \delta \tag{5}$$

We replace $V_C^{\pi_{\phi_{old}}}$ by its sample estimate $\hat{V}_C^{\pi_{\phi_{old}}}$ and denote $\chi - V_C^{\pi_{\phi_{old}}}$ as χ'. Note that χ' is independent of parameter ϕ that is being optimized over. For notational convenience let λ' denote the fraction $\frac{\lambda}{1-\gamma}$, and define $\hat{A}_R^{\pi_{\phi_{old}}} = A_R^{\pi_{\phi_{old}}} - \lambda' A_C$. In addition, we can approximate D_{KL} in terms of the Fisher Information Matrix $\boldsymbol{F}$, where, $\boldsymbol{F}$ can be estimated with samples as

$$\boldsymbol{F} = \mathbb{E}_{s\sim\rho_{\phi_{old}}}\left[\mathbb{E}_{a\sim\pi_{\phi_{old}}}\left[\nabla_{\phi_{old}} \log \pi_{\phi_{old}} (\nabla_{\phi_{old}} \log \pi_{\phi_{old}})^T\right]\right] \tag{6}$$

Following the steps in the Appendix, we can write the **gradient ascent step for** ϕ as

$$\phi \leftarrow \phi_{old} + \beta \boldsymbol{F}^{-1} \nabla_{\phi_{old}} \tilde{J}(\phi_{old}) \quad \beta = \beta^j \sqrt{\frac{2\delta}{\nabla_{\phi_{old}} \tilde{J}(\phi_{old})^T \boldsymbol{F} \nabla_{\phi_{old}} \tilde{J}(\phi_{old})}} \tag{7}$$

Here β^j is the backtracking coefficient and we perform backtracking line search with exponential decay. $\nabla_{\phi_{old}} \tilde{J}(\phi_{old})$ is calculated as,

$$\nabla_{\phi_{old}} \tilde{J}(\phi_{old}) = \mathbb{E}_{s\sim\rho_{\phi_{old}}, a\sim\pi_{\phi_{old}}}\left[\nabla_{\phi_{old}} \log \pi_{\phi_{old}}(a|s) \tilde{A}_R^{\pi_{\phi_{old}}}\right] \tag{8}$$

For **gradient descent with respect to the Lagrange multiplier** λ we have,

$$\lambda \leftarrow \lambda - \eta_\lambda \left(\frac{1}{1-\gamma}\mathbb{E}_{s\sim\rho_{\phi_{old}}, a\sim\pi_{\phi_{old}}}[A_C(s,a)] - \chi'\right) \tag{9}$$

η_λ is the learning rate. Detailed derivations of the gradient updates are in section A.2 of the Appendix.

Executing rollouts (i.e., safe exploration). Since we are interested in minimizing the number of constraint violations while exploring the environment, we do not simply execute the learned policy iterate in the environment for active data collection. Rather, we query the safety critic Q_C to obtain an estimate of how unsafe an action is and choose an actino that is safe via rejection sampling. Formally, we sample an action $a \sim \pi_{\phi_{old}}(s)$, and check if $Q_C(s,a) \leq \epsilon$. We keep re-sampling actions $\pi_{\phi_{old}}(s)$ until this condition is met, and once met, we execute that action in the environment. Here, ϵ is a threshold that varies across iterations and is defined as $\epsilon = (1-\gamma)(\chi - \hat{V}_C^{\pi_{\phi_{old}}}(\mu))$ where, $\hat{V}_C^{\pi_{\phi_{old}}}(\mu)$ is the average episodic failures in the *previous* epoch.

In the replay buffer $\mathcal{D}_{env}$, we store tuples of the form (s, a, s', r, c), where s is the previous state, a is the action executed, s' is the next state, r is the task reward from the environment, and $c = C(s')$, the constraint value. In our setting, c is binary, with 0 denoting a *live* agent and 1 denoting *failure*.

Overall algorithm. Our overall algorithm executes policy rollouts in the environment by respecting the constraint $Q_C(s,a) \leq \epsilon$, stores the observed data tuples in the replay buffer $\mathcal{D}_{env}$, and uses the collected tuples to train a safety value function Q_C using equation 38, update the policy using equation 7, and update the dual variable λ using equation 9.

4 THEORETICAL ANALYSIS

In this section, we aim to theoretically analyze our approach and provably guarantee that the expected probability of failure is bounded at every iteration throughout the learning process, while ensuring that the convergence rate to the optimal solution is only mildly bottlenecked by the additional safety constraint. Our main result, stated in Theorem 1, provides safety guarantee throughout training, by bounding the expected probability of failure of the resulting policy following Equation 4. To prove this, we first state a Lemma that shows that the constraint in Equation 4 are satisfied with high probability during the policy updates. Detailed proofs of all the Lemmas and Theorems are in section A.1 of the Appendix.

Additional notations. Let $\epsilon_C = \max_s |\mathbb{E}_{a\sim\pi_{\phi_{new}}} A_C(s,a)|$ and Δ be the overestimation in $\mathbb{E}_{s\sim\rho_{\phi_{old'}},a\sim\pi_{\phi_{old}}}[A_C(s,a)]$ due to CQL, i.e. $\Delta = \mathbb{E}_{s\sim\rho_{\phi_{old'}},a\sim\pi_{\phi_{old}}}[\hat{A}_C(s,a) - A_C(s,a)]$. Let ζ capture the sampling error in the estimation of $V_C^{\pi_{\phi_{old}}}(\mu)$ and N be the number of samples used in the estimation of V_C.

Lemma 1. *If we follow Algorithm 1, during policy updates via equation 4, the following is satisfied with high probability* $\geq 1-\eta$

$$V_C^{\pi_{\phi_{old}}}(\mu) + \frac{1}{1-\gamma}\mathbb{E}_{s\sim\rho_{\phi_{old}},a\sim\pi_\phi}\left[A_C(s,a)\right] \leq \chi + \zeta - \frac{\Delta}{1-\gamma}$$

Here, ζ captures sampling error in the estimation of $V_C^{\pi_{\phi_{old}}}(\mu)$ and we have $\zeta \leq \frac{C\sqrt{\log(1/\eta)}}{|N|}$, where C is a constant and N is the number of samples used in the estimation of V_C.

We now show that Algorithm 1 provides safety guarantee throughout training by ensuring that the expected probability of failure is bounded.

Theorem 1. *Consider policy updates that solve the constrained optimization problem defined in equation 4. With high probability $\geq 1-\omega$, we have the following upper bound on expected probability of failure $V_C^{\pi_{\phi_{new}}}(\mu)$ for $\pi_{\phi_{new}}$ during every policy update iteration*

$$V_C^{\pi_{\phi_{new}}}(\mu) \leq \chi + \zeta - \frac{\Delta}{1-\gamma} + \frac{\sqrt{2\delta}\gamma\epsilon_C}{(1-\gamma)^2} \qquad \textit{where} \qquad \zeta \leq \frac{C\sqrt{\log(1/\omega)}}{|N|} \tag{10}$$

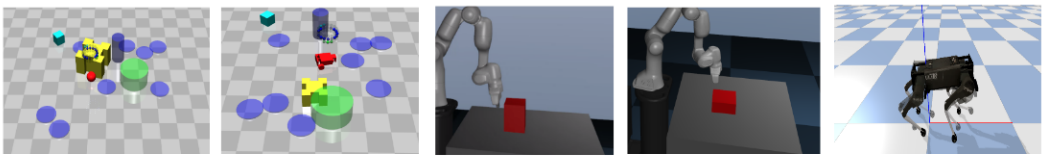
Since ϵ_C depends on the optimized policy $\pi_{\phi_{new}}$, it can't be calculated exactly prior to the update. As we cap $Q_C(s,a)$ to be ≤ 1, therefore, the best bound we can construct for ϵ_C is the trivial bound $\epsilon_C \leq 2$. Now, in order to have $V_C^{\pi_{\phi_{new}}}(\mu) < \chi$, we require $\Delta > \frac{2\sqrt{2\delta}\gamma}{1-\gamma} + \zeta$. To guarantee this, we can obtain a lower bound on α as shown in the proof in section A.1.

So far we have shown that with a high probability we can satisfy the constraint in the objective while updating the current policy $\pi_{\phi_{old}}$ (Lemma 1) and obtain an upper bound on the expected probability of failure of the updated policy $\pi_{\phi_{new}}$ (Theorem 1). We now show that incorporating and satisfying safety constraints during learning does not severely affect the convergence rate to the optimal solution for task performance.

Theorem 2 (Convergence rate for policy gradient updates with the safety constraint)**.** *If we run the policy gradient updates, with a softmax parameterized policy π_ϕ, with μ as the starting state distribution, with $\phi^{(0)} = 0$, and learning rate $\eta_\phi > 0$. For all policy update iterations $T > 0$, we have, with probability $\geq 1-\omega$:*

$$V_R^*(\mu) - V_R^{(T)}(\mu) \leq \frac{\log|\mathcal{A}|}{\eta_\phi T} + \frac{1}{(1-\gamma)^2 T} + ((1-\chi) + (1 - \frac{2\Delta}{(1-\gamma)}) + 2\zeta)\frac{\sum_{t=0}^{T-1}\lambda^{(t)}}{\eta_\phi T}$$

Since, the value of the dual λ only strictly decreases during gradient descent updates (Algorithm 1), $\sum_{t=0}^{T-1}\lambda^{(t)}$ is upper-bounded. In addition, if we choose α as mentioned in Theorem 1, we have



(a) Pointmass (b) Car (c) Panda (d) Panda (e) Laikago

Figure 2: Rendered images of the five environments in our experiments. (a) Pointmass navigation avoiding traps. (b) Car navigation avoiding traps. (c) Panda push without toppling. (d) Panda push within boundary. (e) Laikago walk without falling.

$\Delta > \frac{2\sqrt{2\delta}\gamma}{1-\gamma} + \zeta$. Hence, with probability $\geq 1-\omega$, we can ensure that

$$V_R^*(\mu) - V_R^{(T)}(\mu) \leq \frac{\log|\mathcal{A}|}{\eta T} + \frac{1}{(1-\gamma)^2 T} + K\frac{\sum_{t=0}^{T-1}\lambda^{(t)}}{\eta T} \quad \text{where} \quad K \leq (1-\chi) + \frac{4\sqrt{2\delta}\gamma}{(1-\gamma)^2}$$

So, we see that the additional term introduced in the convergence rate (compared to (Agarwal et al., 2019)) due to the safety constraint is upper bounded, and can be made small with a high probability by choosing α appropriately, even after accounting for sampling error. We discuss some practical considerations of the theoretical results sn Section A.4 of the Appendix.

5 Experiments

Through experiments on challenging continuous control environments, we aim to empirically verify the theoretical guarantees and understand the following questions:

- How *safe* is the proposed approach in terms of constraint satisfaction during training?
- How does learning of *safe policies* trade-off with task performance during training?

5.1 Experimental Setup

Environments. We perform our experimental evaluation across environments with varying difficulty and characteristics. In each environment, we define a task objective that the agent must achieve and a criteria for *catastrophic failure*. The overall objective is to solve the task without dying in the episode. In the ***point agent/car navigation avoiding traps***, the agent must be controlled to navigate a maze, by avoiding traps. The agent has a health counter reset at the beginning of every every episode that decreases for every timestep it spends within a trap. When the counter hits 0, the agent gets *trapped* and *dies*. In ***Panda push without toppling***, a Franka Emika Panda arm must be controlled to push a vertically placed block across the table to a goal location without the block toppling over. *Failure* is defined as when the block topples. In ***Panda push within boundary***, the Panda arm must be controlled to push a block across the table to a goal location without the block going outside a rectangular boundary specified on the table. *failure* is defined as when the block center of mass ((x, y) position) goes outside the boundary during the episode. In ***Laikago walk without falling***, a Laikago quadruped robot must be controlled to walk without falling down. The agent is rewarded for walking as fast as possible (or trotting) and *failure* is defined as when the agent topples over.

Baselines. We compare CSC's three prior methods for a head-to-head comparison: constrained policy optimization (***CPO***) (Achiam et al., 2017), a standard unconstrained RL method which we call ***Base***, and a method with an ensemble of Q-functions similar to safety critics, which we refer to as ***Q ensembles***. The ensembles baseline is an alternate approach for safe exploration that keeps track of epistemic uncertainty, motivated by previous works Eysenbach et al. (2017). There are other safe RL approaches which we cannot compare against, as they make multiple additional assumptions, such as the availability of a function that can be queried to determine if a state is safe or not Thananjeyan et al. (2020), availability of a default safe policy for the task Koller et al. (2018); Berkenkamp et al. (2017), and prior knowledge of the location of unsafe states (Fisac et al., 2019). In addition to the baselines, we analyze variants of our algorithm with different safety thresholds through ablation studies.

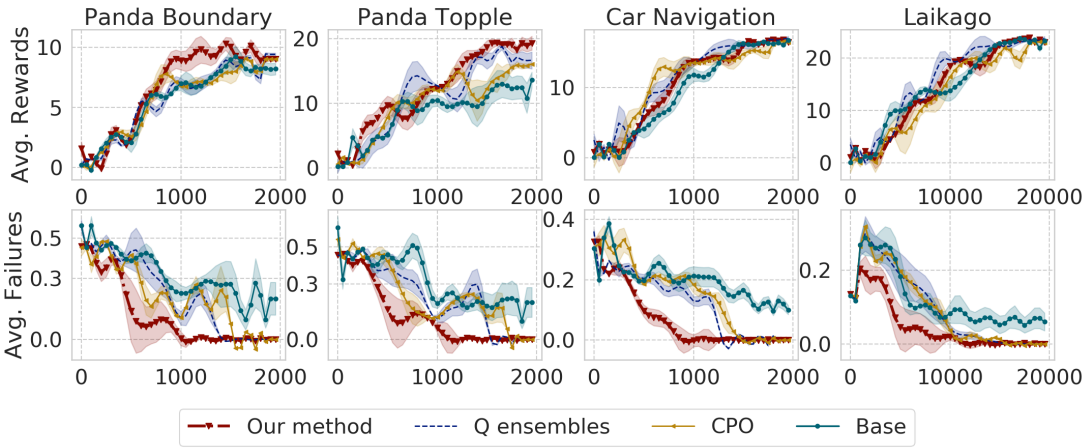


Figure 3: **Top row:** Average task rewards (higher is better). **Bottom row:** Average catastrophic failures (lower is better). Results on four of the five environments we consider for our experiments. For each environment we plot the average task reward, the average episodic failures, and the cumulative episodic failures. Since Laikago is an *extremely* challenging task, for all the baselines, we initialize the agent's policy with a controller that has been trained to keep the agent standing, while not in motion. The task then is to bootstrap learning so that the agent is able to remain standing while walking as well. The safety threshold $\chi = 0.02$ for all the baselines in all the environments. Detailed results including plots of cumulative failures are in Fig. 6 of the Appendix.

5.2 Empirical Results

Comparable or better performance with significantly lower failures during training. In Figure 3, observe that our method has significantly lower average failures per episode and hence lower cumulative failures during the entire training process. Although the failures are lower for our method, we observe that task performance and convergence of average task rewards is competitive and, in some environments, better than prior methods.

CSC naturally trades off performance with safety guarantees, based on the safety-threshold χ. In Figure 4, we plot variants of our method with different safety constraint thresholds χ. Observe that: (a) when the threshold is set to a lower value (stricter constraint), the number of avg. failures per episode decreases in all the environments, and (b) the convergence rate of the task reward is lower when the safety threshold is stricter. These observations empirically complement our theoretical guarantees in Theorems 1 and 2.

6 Related Work

We discuss prior safe RL and safe control methods under three subheadings. We provide a more detailed discussion of related works in section A.7 of the Appendix.

Assuming some prior/domain knowledge of the problem structure. Prior works have attempted to solve safe exploration in the presence of structural assumptions about the environment or safety structures. For example, Koller et al. (2018); Berkenkamp et al. (2017) assume access to a safe set of environment states, and a default safe policy, while in Fisac et al. (2018); Dean et al. (2019), knowledge of system dynamics is assumed and (Fisac et al., 2019) assume access to a distance metric on the state space. SAVED (Thananjeyan et al., 2020) learns a kernel density estimate over unsafe states, and assumes access to a set of user demonstrations and a user specified function that can be queried to determine whether a state is safe or not. In contrast to these approaches, our method does not assume any prior knowledge from the user, or domain knowledge of the problem setting, except a binary signal from the environment indicating when a catastrophic failure has occurred.

Assuming a continuous safety cost function. CPO (Achiam et al., 2017) assumes a cost function can be queried from the environment at every time-step and the objective is to keep the cumulative costs within a certain limit. This assumption limits the generality of the method in scenarios where

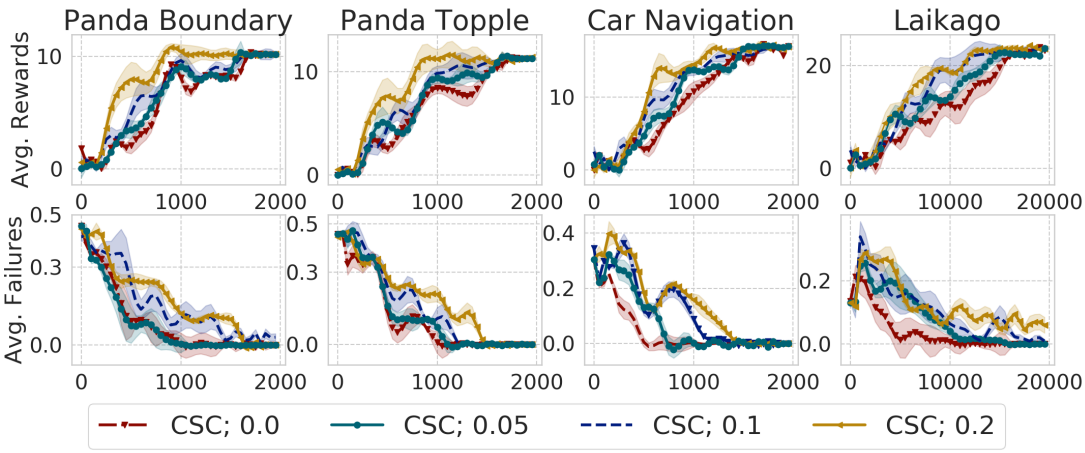


Figure 4: **Top row:** Average task rewards (higher is better). **Bottom row:** Average catastrophic failures (lower is better). Results on four of the five environments we consider for our experiments. For each environment we plot the average task reward, the average episodic failures, and the cumulative episodic failures. All the plots are for our method (CSC) with different safety thresholds χ, specified in the legend. From the plots it is evident that our method can naturally trade-off safety for task performance depending on how strict the safety threshold is set to. Detailed results including plots of cumulative failures are in Fig. 5 of the Appendix.

only minimal feedback, such as binary reward feedback is provided. (Stooke et al., 2020) devise a general modification to the Lagrangian by incorporating two additional terms in the optimization of the dual variable. SAMBA (Cowen-Rivers et al., 2020) has a learned GP dynamics model and a continuous constraint cost function that encodes safety. The objective is to minimize task cost function while maintaining the CVAR_α of cumulative costs below a threshold. In Dalal et al. (2018); Paternain et al. (2019b;a); Grbic & Risi (2020), only the optimal policy is learned to be safe, and there are no safety guarantees during training. In contrast to these approaches, we assume only a binary signal from the environment indicating when a catastrophic failure has occurred. Instead of minimizing expected costs, our constraint formulation directly seeks to constrain the expected probability of failure.

Safety through reversibility/recoverability. Prior works have attempted to devise resetting mechanisms to recover the policy to a base configuration from (near) a potentially unsafe state. LNT (Eysenbach et al., 2017) trains both a forward policy for solving a task, and a reset goal-conditioned policy that kicks in when the agent is in an *unsafe* state and learns an ensemble of critics, which is substantially more complex than our approach of a learned safety critic, which can give rise to a simple but provable safe exploration algorithm. In control theory, a number of prior works have focused on Hamilton-Jacobi-Isaacs (HJI) reachability analysis (Bansal et al., 2017) for providing safety guarantees and obtaining control inputs for dynamical systems (Herbert et al., 2019; Bajcsy et al., 2019; Leung et al., 2018). Our method does not require knowledge of the system dynamics or regularity conditions on the state-space, which are crucial to be able to compute unsafe states using HJI reachability.

7 Discussion and Conclusion

We described a safe exploration algorithm that learns a conservative estimate of the safety value of every state, and uses this to constraint policy evaluation and policy improvement. We provably demonstrated constraint upper bound satisfaction and convergence guarantees of the proposed algorithm. We empirically validated our theoretical guarantees and showed that the proposed algorithm achieves high task performance while incurring significantly low accidents during training.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. *arXiv preprint arXiv:1705.10528*, 2017.

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in markov decision processes. *arXiv preprint arXiv:1908.00261*, 2019.

Eitan Altman. *Constrained Markov decision processes*, volume 7. CRC Press, 1999.

Andrea Bajcsy, Somil Bansal, Eli Bronstein, Varun Tolani, and Claire J Tomlin. An efficient reachability-based framework for provably safe autonomous navigation in unknown environments. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 1758–1765. IEEE, 2019.

Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J Tomlin. Hamilton-jacobi reachability: A brief overview and recent advances. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 2242–2253. IEEE, 2017.

Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. In *Advances in neural information processing systems*, pp. 908–918, 2017.

Alexander I Cowen-Rivers, Daniel Palenicek, Vincent Moens, Mohammed Abdullah, Aivar Sootla, Jun Wang, and Haitham Ammar. Samba: Safe model-based & active reinforcement learning. *arXiv preprint arXiv:2006.09436*, 2020.

Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. Safe exploration in continuous action spaces. *arXiv preprint arXiv:1801.08757*, 2018.

Sarah Dean, Stephen Tu, Nikolai Matni, and Benjamin Recht. Safely learning to control the constrained linear quadratic regulator. In *2019 American Control Conference (ACC)*, pp. 5582–5588. IEEE, 2019.

Benjamin Eysenbach, Shixiang Gu, Julian Ibarz, and Sergey Levine. Leave no trace: Learning to reset for safe and autonomous reinforcement learning. *arXiv preprint arXiv:1711.06782*, 2017.

Jaime F Fisac, Anayo K Akametalu, Melanie N Zeilinger, Shahab Kaynama, Jeremy Gillula, and Claire J Tomlin. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Transactions on Automatic Control*, 64(7):2737–2752, 2018.

Jaime F Fisac, Neil F Lugovoy, Vicenç Rubies-Royo, Shromona Ghosh, and Claire J Tomlin. Bridging hamilton-jacobi safety analysis and reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8550–8556. IEEE, 2019.

Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Djordje Grbic and Sebastian Risi. Safe reinforcement learning through meta-learned instincts. *arXiv preprint arXiv:2005.03233*, 2020.

Sylvia L Herbert, Somil Bansal, Shromona Ghosh, and Claire J Tomlin. Reachability-based safety guarantees using efficient initializations. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 4810–4816. IEEE, 2019.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *ICML*, volume 2, pp. 267–274, 2002.

Torsten Koller, Felix Berkenkamp, Matteo Turchetta, and Andreas Krause. Learning-based model predictive control for safe exploration. In *2018 IEEE Conference on Decision and Control (CDC)*, pp. 6059–6066. IEEE, 2018.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *arXiv preprint arXiv:2006.04779*, 2020.

Sascha Lange, Thomas Gabel, and Martin Riedmiller. Batch reinforcement learning. In *Reinforcement learning*, pp. 45–73. Springer, 2012.

Karen Leung, Edward Schmerling, Mo Chen, John Talbot, J Christian Gerdes, and Marco Pavone. On infusing reachability-based safety assurance within probabilistic planning frameworks for human-robot vehicle interactions. In *International Symposium on Experimental Robotics*, pp. 561–574. Springer, 2018.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Santiago Paternain, Miguel Calvo-Fullana, Luiz FO Chamon, and Alejandro Ribeiro. Safe policies for reinforcement learning via primal-dual methods. *arXiv preprint arXiv:1911.09101*, 2019a.

Santiago Paternain, Luiz Chamon, Miguel Calvo-Fullana, and Alejandro Ribeiro. Constrained reinforcement learning has zero duality gap. In *Advances in Neural Information Processing Systems*, pp. 7555–7565, 2019b.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897, 2015.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by pid lagrangian methods. *arXiv preprint arXiv:2007.03964*, 2020.

Chen Tessler, Daniel J Mankowitz, and Shie Mannor. Reward constrained policy optimization. *arXiv preprint arXiv:1805.11074*, 2018.

Brijen Thananjeyan, Ashwin Balakrishna, Ugo Rosolia, Felix Li, Rowan McAllister, Joseph E Gonzalez, Sergey Levine, Francesco Borrelli, and Ken Goldberg. Safety augmented value estimation from demonstrations (saved): Safe deep model-based rl for sparse cost robotic tasks. *IEEE Robotics and Automation Letters*, 5(2):3612–3619, 2020.