Accelerating Safe Reinforcement Learning with Constraint-mismatched Baseline Policies

Tsung-Yen Yang [1] **Justinian Rosca** [2] **Karthik Narasimhan** [1] **Peter J. Ramadge** [1]

Abstract

We consider the problem of reinforcement learning when provided with (1) *a baseline control policy* and (2) *a set of constraints* that the learner must satisfy. The baseline policy can arise from demonstration data or a teacher agent and may provide useful cues for learning, but it might also be sub-optimal for the task at hand, and is not guaranteed to satisfy the specified constraints, which might encode safety, fairness or other application-specific requirements. In order to safely learn from baseline policies, we propose an iterative policy optimization algorithm that alternates between maximizing expected return on the task, minimizing distance to the baseline policy, and projecting the policy onto the constraint-satisfying set. We analyze our algorithm theoretically and provide a finite-time convergence guarantee. In our experiments on five different control tasks, our algorithm consistently outperforms several state-of-the-art baselines, achieving 10 times fewer constraint violations and 40% higher reward on average.

1. Introduction

Deep reinforcement learning (RL) has achieved impressive results in several domains such as games (Mnih et al., 2013; Silver et al., 2016) and robotic control (Levine et al., 2016; Rajeswaran et al., 2017). However, in these complex applications, learning policies from scratch often requires tremendous amounts of time and computational power. To alleviate this issue, one would like to leverage a baseline policy available from demonstrations, a teacher or a previous task. However, the baseline policy may be sub-optimal for the new application and may not be guaranteed to produce actions that satisfy desired constraints on safety, fairness, or other costs. For instance, when you drive an unfamiliar vehicle, you do so cautiously to ensure safety, while adapting your driving technique to the vehicle characteristics to improve your 'driving reward'. In effect, you (as the agent) gradually adapt a baseline policy (*i.e.,* prior driving skill) to avoid violating the constraints (*e.g.,* safety) while improving your driving reward (*e.g.,* travel time, fuel efficiency).

The problem of safely learning from baseline policies is challenging because directly leveraging the baseline policy, as in DAGGER (Ross et al., 2011) or GAIL (Ho & Ermon, 2016), may result in policies that violate the constraints since the baseline is not guaranteed to satisfy them. To ensure constraint satisfaction, prior work either adds a hyper-parameter weighted copy of the imitation learning (IL) objective (*i.e.,* imitating the baseline policy) to the RL objective (Rajeswaran et al., 2017; Gao et al., 2018; Hester et al., 2018), or pre-trains a policy with the baseline policy (*e.g.,* use a baseline policy as an initial policy) and then fine-tunes it through RL (Mülling et al., 2013; Chernova & Thomaz, 2014). However, both approaches do not ensure constraint satisfaction on *every* learning episode, which is an important feature of safe RL. In addition, the policy initialized by a low entropy baseline policy may never explore.

In this work, to learn from the baseline policy while satisfying constraints, we propose an iterative algorithm that performs policy updates in three stages. The first step updates the policy to maximize expected reward using trust region policy optimization (*e.g.,* TRPO (Schulman et al., 2015)). This can, however, result in a new intermediate policy that is too far from the baseline policy and may not satisfy the constraints. The second step performs a projection in policy space to control the distance between the current policy and the baseline policy. In contrast to the approach that regularizes the standard RL objective with the distance w.r.t. the baseline policy and makes the regularization parameter fade over time, our approach allows the learning agent to update the distance when needed. In addition, this step allows the agent to explore without being overly restricted by the potentially constraint-violating baseline policy. This also enables the baseline policy to influence the learning even at later iterations without the computational burden of learning a cost function for the baseline policy (Kwon et al., 2020). The third step ensures constraint satisfaction at every

[1]Princeton University [2]Siemens Corporation, Corporate Technology. Correspondence to: Tsung-Yen Yang <ty3@princeton.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

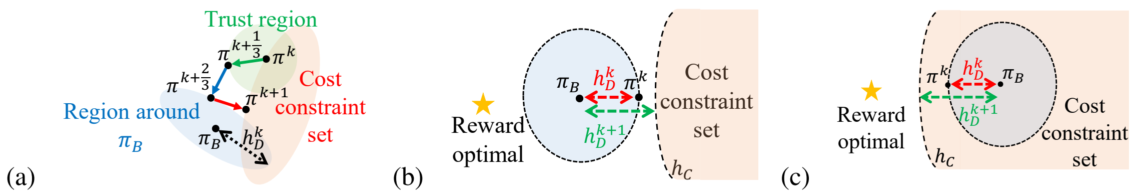


Figure 1. **(a)** Update procedures for SPACE. Step 1 (green) improves the reward in the trust region. Step 2 (blue) projects the policy onto an *adaptable* region around the baseline policy π_B. Step 3 (red) projects the policy onto the constraint set. **(b)** Illustrating when π_B is *outside* the constraint set. **(c)** Illustrating when π_B is *inside* the constraint set. The highest reward is achieved at the yellow star. h^k_D (the distance between π^k and π_B) is updated to h^{k+1}_D to ensure constraint satisfaction and exploration of the agent.

iteration by performing a projection onto the set of policies that satisfy the given constraints. We call our algorithm *Safe Policy Adaptation with Constrained Exploration* (SPACE).

This paper's contributions are two-fold. **(1)** We explicitly examine how the baseline policy affects the cost violations of the agent and hence provide a method to safely learn from the baseline policy. This is done by controlling the distance between the learned policy at iteration k and the baseline policy to ensure both feasibility of the optimization problem and safe exploration by the learning agent (Fig. 1(b) and (c)). Such approach, in contrast to non-adaptable constraint sets and learning a policy from scratch (Yang et al., 2020), leads to better sample efficiency and hence are more favorable in real applications. To our knowledge, prior work does not carry out such analysis. We further provide a finite-time guarantee for the convergence of SPACE. **(2)** Second, we empirically show that SPACE can robustly learn from sub-optimal baseline policies in a diverse set of tasks. These include two Mujoco tasks with safety constraints, and two real-world traffic management tasks with fairness constraints. We further show that our algorithm can safely learn from a *human demonstration* driving policy with safety constraints. In all cases, SPACE outperforms state-of-the-art safe RL algorithms, averaging 40% more reward with 10 times fewer cost violations. This shows that SPACE safely and efficiently leverages the baseline policy, and represents a step towards safe deployment of RL in real applications[1].

2. Related Work

Safe RL. Learning constraint-satisfying policies has been explored in the context of safe RL (Garcia & Fernandez, 2015; Hasanbeig et al., 2020; Junges et al., 2016; Jansen et al., 2020; Chow et al., 2018; Bharadhwaj et al., 2020; Srinivasan et al., 2020). Prior work either uses a conditional-gradient approach (Achiam et al., 2017), adds a weighted copy of the cost objective in the reward function (Tessler et al., 2019; Chow et al., 2019; Fujimoto et al., 2019; Stooke et al., 2020), adds a safety layer to the policy (Dalal et al., 2018; Avni et al., 2019), or uses the chanced constraints (Fu & Prashanth L, 2018; Zheng & Ratliff, 2020). In contrast, we use projections to ensure safety.

In addition, Thananjeyan et al. (2021) use the sub-optimal demonstration (still safe π_B) to guide the learning. They obtain the safe policy by iteratively solving model predictive control. However, we focus on the model-free setting, which makes it hard to compare to our method. While Zhang et al. (2020); Srinivasan et al. (2020); Thananjeyan et al. (2020), pre-train a safe policy, they do not focus on how to safely use baseline policies. Moreover, we do not have two stages of pre-training and fine-tuning.

Yang et al. (2020) also uses projections to ensure safety–Projection-based Constrained Policy Optimization (PCPO). However, we show that treating learning from the baseline policy as another fixed constraint in PCPO results in cost constraint violations or sub-optimal reward performance. Instead, our main idea is to have an *adaptable* constraint set that adjusts the distance between the baseline and the learning policies at each iteration with the distance controlled by the learning progress of the agent, *i.e.,* the reward improvement and the cost constraint violations. Such approach ensures exploration and cost satisfaction of the agent. Please refer to Section 5 for the detailed comparison to PCPO.

Policy optimization with the initial safe set. Wachi & Sui (2020); Sui et al. (2015); Turchetta et al. (2016) assume that the initial safe set is given, and the agent explores the environment and verifies the safety function from this initial safe set. In contrast, our assumption is to give a baseline policy to the agent. Both assumptions are reasonable as they provide an initial understanding of the environment.

Leveraging baseline policies for RL. Prior work has used baseline policies to provide initial information to RL algorithms to reduce or avoid undesirable situations. This is done by either: initializing the policy with the baseline policy (Driessens & Džeroski, 2004; Smart & Kaelbling, 2000; Koppejan & Whiteson, 2011; Abbeel et al., 2010; Gao et al., 2018; Le et al., 2019; Vecerik et al., 2017; Jaques et al., 2019), or providing a teacher's advice to the agent (Garcia & Fernández, 2012; Quintía Vidal et al., 2013; Abel et al.,

[1]Code is available at: `https://sites.google.com/view/spacealgo`

2017; Zhang et al., 2019). However, such works often assume that the baseline policy is constraint-satisfying (Sun et al., 2018; Balakrishna et al., 2019). In contrast, SPACE safely leverages the baseline policy without requiring it to satisfy the specified constraints. Pathak et al. (2015); Bartocci et al. (2011) also modify the existing known models (policies) based on new conditions in the context of the formal methods. In contrast, we solve this problem using projections in the policy space.

Learning from logged demonstration data. To effectively learn from demonstration data given by the baseline policy, Wu et al. (2019); Brown et al. (2019); Kwon et al. (2020) assess the demonstration data by either: predicting their cost in the new task using generative adversarial networks (GANs) (Goodfellow et al., 2014), or directly learning the cost function of the demonstration data. In contrast, SPACE controls the distance between the learned and baseline policies to ensure learning improvement.

3. Problem Formulation

We frame our problem as a constrained Markov Decision Process (CMDP) (Altman, 1999), defined as a tuple $< \mathcal{S}, \mathcal{A}, T, R, C >$. Here $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, and T specifies the conditional probability $T(s'|s,a)$ that the next state is s' given the current state s and action a. In addition, $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is a reward function, and $C : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is a constraint cost function. The reward function encodes the benefit of using action a in state s, while the cost function encodes the corresponding constraint violation penalty.

A policy is a map from states to probability distributions on $\mathcal{A}$. It specifies that in state s the selected action is drawn from the distribution $\pi(s)$. The state then transits from s to s' according to the state transition distribution $T(s'|s,a)$. In doing so, a reward $R(s,a)$ is received and a constraint cost $C(s,a)$ is incurred, as outlined above.

Let $\gamma \in (0,1)$ denote a discount factor, and τ denote the trajectory $\tau = (s_0, a_0, s_1, \cdots)$ induced by a policy π. Normally, we seek a policy π that maximizes a cumulative discounted reward

$$J_R(\pi) \doteq \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t,a_t)\right], \tag{1}$$

while keeping the cumulative discounted cost below h_C

$$J_C(\pi) \doteq \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t C(s_t,a_t)\right] \leq h_C. \tag{2}$$

Here we consider an additional objective. We are provided with a baseline policy π_B and at each state s we measure the divergence between $\pi(s)$ and $\pi_B(s)$. For example, this could be the KL-divergence $D(s) \doteq D_{\mathrm{KL}}(\pi(s)\|\pi_B(s))$. We then seek a policy that maximizes Eq. (1), satisfies Eq. (2), and ensures the discounted divergence between the learned and baseline policies is below h_D:

$$J_D(\pi) \doteq \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t D(s_t)\right] \leq h_D. \tag{3}$$

We do not assume that the baseline policy satisfies the cost constraint. Hence we allow h_D to be adjusted during the learning of π to allow for reward improvement and constraint satisfaction.

Let $\mu_t(\cdot|\pi)$ denote the state distribution at time t under policy π. The discounted state distribution induced by π is defined to be $d^\pi(s) \doteq (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\mu_t(s|\pi)$. Now bring in the reward advantage function (Kakade & Langford, 2002) defined by

$$A_R^\pi(s,a) \doteq Q_R^\pi(s,a) - V_R^\pi(s),$$

where $V_R^\pi(s) \doteq \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t,a_t)|s_0=s\right]$ is the expected reward from state s under policy π, and $Q_R^\pi(s,a) \doteq \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t,a_t)|s_0=s, a_0=a\right]$ is the expected reward from state s and initial action a, and thereafter following policy π. These definitions allow us to express the reward performance of one policy π' in terms of another π:

$$J_R(\pi') - J_R(\pi) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi'}, a\sim\pi'}[A_R^\pi(s,a)].$$

Similarly, we can define $A_D^\pi(s,a)$, $Q_D^\pi(s,a)$ and $V_D^\pi(s)$ for the divergence cost, and $A_C^\pi(s,a)$, $Q_C^\pi(s,a)$ and $V_C^\pi(s)$ for the constraint cost.

4. Safe Policy Adaptation with Constrained Exploration (SPACE)

We now describe the proposed iterative algorithm illustrated in Fig. 1. In what follows, π^k denotes the learned policy after iteration k, and M denotes a distance measure between policies. For example, M may be the 2-norm of the difference of policy parameters or some average over the states of the KL-divergence of the action policy distributions.

Step 1. We perform one step of trust region policy optimization (Schulman et al., 2015). This maximizes the reward advantage function $A_R^\pi(s,a)$ over a KL-divergence neighborhood of π^k:

$$\begin{aligned}\pi^{k+\frac{1}{3}} = \arg\max_{\pi}\ & \mathbb{E}_{s\sim d^{\pi^k},\, a\sim\pi}[A_R^{\pi^k}(s,a)] \\ \text{s.t. } & \mathbb{E}_{s\sim d^{\pi^k}}\left[D_{\mathrm{KL}}(\pi(s)\|\pi^k(s))\right] \leq \delta.\end{aligned} \tag{4}$$

Step 2. We project $\pi^{k+\frac{1}{3}}$ onto a region around π_B controlled by h_D^k to minimize M:

$$\begin{aligned}\pi^{k+\frac{2}{3}} &= \arg\min_{\pi}\ M(\pi, \pi^{k+\frac{1}{3}})\\ \text{s.t. } J_D(\pi^k) &+ \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi^k},\, a\sim\pi}[A_D^{\pi^k}(s)] \leq h_D^k.\end{aligned} \tag{5}$$

Step 3. We project $\pi^{k+\frac{2}{3}}$ onto the set of policies satisfying the cost constraint to minimize M:

$$\begin{aligned}\pi^{k+1} &= \arg\min_{\pi}\ M(\pi, \pi^{k+\frac{2}{3}})\\ \text{s.t. } J_C(\pi^k) &+ \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi^k},\, a\sim\pi}[A_C^{\pi^k}(s,a)] \leq h_C.\end{aligned} \tag{6}$$

Remarks. Since we use a small step size δ, we can replace the state distribution d^π with d^{π^k} in Eq. (5) and (6) and hence compute $A_D^{\pi^k}$ and $A_C^{\pi^k}$. Please see the supplementary material for the derivation of this approximation.

Control h_D^k in Step 2. We select h_D^0 to be small and gradually increase h_D^k at each iteration to expand the region around π_B. Specifically, we make $h_D^{k+1} > h_D^k$ if:

(a) $J_C(\pi^k) > J_C(\pi^{k-1})$: this increase is to ensure a nonempty intersection between the region around π_B and the cost constraint set (feasibility). See Fig. 1(b).

(b) $J_R(\pi^k) < J_R(\pi^{k-1})$: this increase gives the next policy more freedom to improve the reward and the cost constraint performance (exploration). See Fig. 1(c).

It remains to determine how to set the new value of h_D^{k+1}. Let $\mathcal{U}_1$ denote the set of policies satisfying the cost constraint, and $\mathcal{U}_2^k$ denote the set of policies in the region around π_B controlled by h_D^k. Then we have the following Lemma.

Lemma 4.1 (**Updating** h_D). *If at step $k+1$: $h_D^{k+1} \geq \mathcal{O}\big((J_C(\pi^k)-h_C)^2\big)+h_D^k$, then $\mathcal{U}_1 \cap \mathcal{U}_2^{k+1} \neq \emptyset$ (feasibility) and $\mathcal{U}_2^{k+1} \cap \partial\mathcal{U}_1 \neq \emptyset$ (exploration).*

Proof. Proved by Three-point Lemma (Chen & Teboulle, 1993). See the supplementary material for more details. □

Remarks. Two values are in the big $\mathcal{O}$. The first value depends on the discounted factor γ, and the second value depends on relative distances between π^k, π_B, and the policy in $\partial\mathcal{U}_1$. The intuition is that the smaller the distances are, the smaller the update of h_D^k is.

Importantly, Lemma 4.1 ensures that the boundaries of the region around π_B determined by h_D and the set of policies satisfying the cost constraint intersect. Note that h_D will become large enough to guarantee feasibility during training. This *adaptable* constraint set, in contrast to the *fixed* constraint set in PCPO, allows the learning algorithm to explore policies within the cost constraint set while still learning from the baseline policy. Compared to other CMDP approaches, the step of projecting close to π_B allows the policy to quickly improve. Compared to behavior cloning, the steps of reward optimization and constraint projection allow the policy to achieve good final performance. We examine the importance of updating h_D in Section 6.

Algorithm 1 SPACE

Initialize a policy $\pi^0 = \pi(\cdot|\boldsymbol{\theta}^0)$ and a trajectory buffer $\mathcal{B}$
for $k = 0, 1, 2, \cdots$ **do**
  Run $\pi^k = \pi(\cdot|\boldsymbol{\theta}^k)$ and store trajectories in $\mathcal{B}$
  Obtain $\boldsymbol{\theta}^{k+1}$ using the update in Eq. (10)
  If $J_C(\pi^k) > J_C(\pi^{k-1})$ or $J_R(\pi^k) < J_R(\pi^{k-1})$
    Update h_D^{k+1} using **Lemma 4.1**
  Empty $\mathcal{B}$

5. A Theoretical Analysis of SPACE

We will implement a policy as a neural network with fixed architecture parameterized by $\boldsymbol{\theta} \in \mathbb{R}^n$. We then learn a policy from the achievable set $\{\pi(\cdot|\boldsymbol{\theta})\colon \boldsymbol{\theta} \in \mathbb{R}^n\}$ by iteratively learning $\boldsymbol{\theta}$. Let $\boldsymbol{\theta}^k$ and $\pi^k \doteq \pi(\cdot|\boldsymbol{\theta}^k)$ denote the parameter value and the corresponding policy at step k. In this setting, it is impractical to solve for the policy updates in Eq. (4), (5) and (6). Hence we approximate the reward function and constraints with first order Taylor expansions, and KL-divergence with a second order Taylor expansion. We will need the following derivatives:

(1) $\boldsymbol{g}^k \doteq \nabla_{\boldsymbol{\theta}}\mathbb{E}_{s\sim d^{\pi^k},\, a\sim\pi}[A_R^{\pi^k}(s,a)]$,

(2) $\boldsymbol{a}^k \doteq \nabla_{\boldsymbol{\theta}}\mathbb{E}_{s\sim d^{\pi^k},\, a\sim\pi}[A_D^{\pi^k}(s)]$,

(3) $\boldsymbol{c}^k \doteq \nabla_{\boldsymbol{\theta}}\mathbb{E}_{s\sim d^{\pi^k},\, a\sim\pi}[A_C^{\pi^k}(s,a)]$, and

(4) $\boldsymbol{F}^k \doteq \nabla^2_{\boldsymbol{\theta}}\mathbb{E}_{s\sim d^{\pi^k}}\left[D_{\mathrm{KL}}(\pi(s)\|\pi^k(s))\right]$.

Each of these derivatives are taken w.r.t. the neural network parameter and evaluated at $\boldsymbol{\theta}^k$. We also define $b^k \doteq J_D(\pi^k) - h_D^k$, and $d^k \doteq J_C(\pi^k) - h_C$. Let $u^k \doteq \sqrt{\frac{2\delta}{\boldsymbol{g}^{k^T}\boldsymbol{F}^{k^{-1}}\boldsymbol{g}^k}}$, and $\boldsymbol{L} = \boldsymbol{I}$ for the 2-norm projection and $\boldsymbol{L} = \boldsymbol{F}^k$ for the KL-divergence projection.

Step 1. Approximating Eq. (4) yields

$$\begin{aligned}\boldsymbol{\theta}^{k+\frac{1}{3}} = \arg\max_{\boldsymbol{\theta}}\ &\boldsymbol{g}^{k^T}(\boldsymbol{\theta}-\boldsymbol{\theta}^k)\\ \text{s.t. } &\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}^k)^T\boldsymbol{F}^k(\boldsymbol{\theta}-\boldsymbol{\theta}^k) \leq \delta.\end{aligned} \tag{7}$$

Step 2 & 3. Approximating Eq. (5) and (6), similarly yields

$$\begin{aligned}\boldsymbol{\theta}^{k+\frac{2}{3}} = \arg\min_{\boldsymbol{\theta}}\ &\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}^{k+\frac{1}{3}})^T\boldsymbol{L}(\boldsymbol{\theta}-\boldsymbol{\theta}^{k+\frac{1}{3}})\\ \text{s.t. } &\boldsymbol{a}^{k^T}(\boldsymbol{\theta}-\boldsymbol{\theta}^k) + b^k \leq 0,\end{aligned} \tag{8}$$

$$\begin{aligned}\boldsymbol{\theta}^{k+1} = \arg\min_{\boldsymbol{\theta}}\ &\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}^{k+\frac{2}{3}})^T\boldsymbol{L}(\boldsymbol{\theta}-\boldsymbol{\theta}^{k+\frac{2}{3}})\\ \text{s.t. } &\boldsymbol{c}^{k^T}(\boldsymbol{\theta}-\boldsymbol{\theta}^k) + d^k \leq 0,\end{aligned} \tag{9}$$

where $\boldsymbol{L} = \boldsymbol{I}$ for the 2-norm projection and $\boldsymbol{L} = \boldsymbol{F}^k$ for the KL-divergence projection. We solve these problems using convex programming, then we have ($(\cdot)^+$ is $\max(0,\cdot)$)

$$\begin{aligned}\boldsymbol{\theta}^{k+1} &= \boldsymbol{\theta}^k + u^k \boldsymbol{F}^{k^{-1}} \boldsymbol{g}^k \\ &- \Big(\frac{u^k \boldsymbol{a}^{k^T} \boldsymbol{F}^{k^{-1}} \boldsymbol{g}^k + b^k}{\boldsymbol{a}^{k^T} \boldsymbol{L}^{-1} \boldsymbol{a}^k}\Big)^+ \boldsymbol{L}^{-1} \boldsymbol{a}^k \\ &- \Big(\frac{u^k \boldsymbol{c}^{k^T} \boldsymbol{F}^{k^{-1}} \boldsymbol{g}^k + d^k}{\boldsymbol{c}^{k^T} \boldsymbol{L}^{-1} \boldsymbol{c}^k}\Big)^+ \boldsymbol{L}^{-1} \boldsymbol{c}^k.\end{aligned} \tag{10}$$

Algorithm 1 shows the corresponding pseudocode.

Convergence analysis. We consider the following simplified problem to provide a convergence guarantee of SPACE:

$$\min_{\boldsymbol{\theta} \in \mathcal{C}_1 \cap \mathcal{C}_2} f(\boldsymbol{\theta}), \tag{11}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a twice continuously differentiable function at every point in a open set $\mathcal{X} \subseteq \mathbb{R}^n$, and $\mathcal{C}_1 \subseteq \mathcal{X}$ and $\mathcal{C}_2 \subseteq \mathcal{X}$ are compact convex sets with $\mathcal{C}_1 \cap \mathcal{C}_2 \neq \emptyset$. The function f is the negative reward function of our CMDP, and the two constraint sets represent the cost constraint set and the region around the baseline policy π_B.

For a vector $\boldsymbol{x}$, let $\|\boldsymbol{x}\|$ denote the Euclidean norm. For a matrix $\boldsymbol{M}$ let $\|\boldsymbol{M}\|$ denote the induced matrix 2-norm, and $\sigma_i(\boldsymbol{M})$ denote the i-th largest singular value of $\boldsymbol{M}$.

Assumption 1. We assume:

(1.1) The gradient ∇f is L-Lipschitz continuous over a open set $\mathcal{X}$.

(1.2) For some constant G, $\|\nabla f(\boldsymbol{\theta})\| \leq G$.

(1.3) For a constant H, $\mathrm{diam}(\mathcal{C}_1) \leq H$ and $\mathrm{diam}(\mathcal{C}_2) \leq H$.

Assumptions (1.1) and (1.2) ensure that the gradient can not change too rapidly and the norm of the gradient can not be too large. (1.3) implies that for every iteration, the diameter of the region around π_B is bounded above by H.

We will need a concept of an ϵ-first order stationary point (Mokhtari et al., 2018). For $\epsilon > 0$, we say that $\boldsymbol{\theta}^* \in \mathcal{C}_1 \cap \mathcal{C}_2$ an ϵ-first order stationary point (ϵ-FOSP) of Problem (11) under KL-divergence projection if

$$\nabla f(\boldsymbol{\theta}^*)^T (\boldsymbol{\theta} - \boldsymbol{\theta}^*) \geq -\epsilon, \quad \forall \boldsymbol{\theta} \in \mathcal{C}_1 \cap \mathcal{C}_2. \tag{12}$$

Similarly, under the 2-norm projection, $\boldsymbol{\theta}^* \in \mathcal{C}_1 \cap \mathcal{C}_2$ an ϵ-FOSP of (11) if

$$\nabla f(\boldsymbol{\theta}^*)^T \boldsymbol{F}^* (\boldsymbol{\theta} - \boldsymbol{\theta}^*) \geq -\epsilon, \quad \forall \boldsymbol{\theta} \in \mathcal{C}_1 \cap \mathcal{C}_2, \tag{13}$$

where $\boldsymbol{F}^* \doteq \nabla^2_{\boldsymbol{\theta}} \mathbb{E}_{s \sim d^{\pi^*}} \left[D_{\mathrm{KL}}(\pi(s) \| \pi^*(s)) \right]$. Notice that SPACE converges to distinct stationary points under the two possible projections (see the supplementary material). With these assumptions, we have the following Theorem.

Theorem 5.1 (Finite-Time Convergence Guarantee of SPACE). *Under the KL-divergence projection, there exists a sequence $\{\eta^k\}$ such that SPACE converges to an ϵ-FOSP in at most $\mathcal{O}(\epsilon^{-2})$ iterations. Moreover, at step $k+1$*

$$f(\boldsymbol{\theta}^{k+1}) \leq f(\boldsymbol{\theta}^k) - \frac{L\epsilon^2}{2(G + \frac{H\sigma_1(\boldsymbol{F}^k)}{\eta^k})^2}. \tag{14}$$

Similarly, under the 2-norm projection, there exists a sequence $\{\eta^k\}$ such that SPACE converges to an ϵ-FOSP in at most $\mathcal{O}(\epsilon^{-2})$ iterations. Moreover, at step $k+1$

$$f(\boldsymbol{\theta}^{k+1}) \leq f(\boldsymbol{\theta}^k) - \frac{L\epsilon^2}{2(G\sigma_1(\boldsymbol{F}^{k^{-1}}) + \frac{H}{\eta^k})^2}. \tag{15}$$

Proof. The proof and the sequence $\{\eta^k\}$ are given in the supplementary material. □

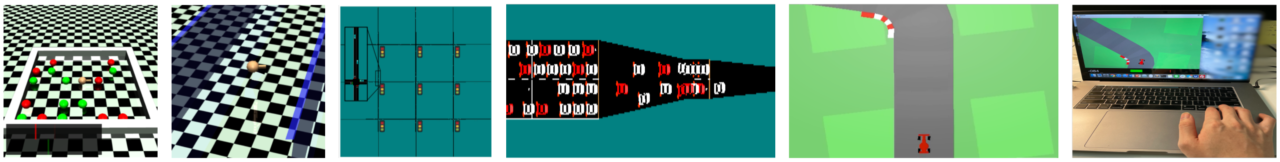
We now make several observations for Theorem 5.1.

(1) The smaller H is, the greater the decrease in the objective. This observation supports the idea of starting with a small value for h_D and increasing it only when needed.

(2) Under the KL-divergence projection, the effect of $\sigma_1(\boldsymbol{F}^k)$ is negligible. This is because in this case η^k is proportional to $\sigma_1(\boldsymbol{F}^k)$. Hence $\sigma_1(\boldsymbol{F}^k)$ does not play a major role in decreasing the objective value.

(3) Under the 2-norm projection, the smaller $\sigma_1(\boldsymbol{F}^{k^{-1}})$ (*i.e.,* larger $\sigma_n(\boldsymbol{F}^k)$) is, the greater the decrease in the objective. This is because a large $\sigma_n(\boldsymbol{F}^k)$ means a large curvature of f in all directions. This implies that the 2-norm distance between the pre-projection and post-projection points is small, leading to a small deviation from the reward improvement direction after doing projections.

Comparison to Yang et al. (2020). Our work is inspired by PCPO (Yang et al., 2020), which also uses projections to ensure constraint satisfaction during policy learning. However, there are a few key differences between our work and PCPO. **(1) Algorithm.** PCPO does not have the capability to safely exploit a baseline policy, which makes it less sample efficient in cases when we have demonstrations or teacher agents. In addition, SPACE's update dynamically sets distances between policies while PCPO does not–this update is important to effectively and safely learn from the baseline policy. **(2) Theory.** Our analysis provides a safety guarantee to ensure the feasibility of the optimization problem while Yang et al. (2020) do not. Merely adding an IL objective in the reward objective of PCPO cannot make the agent learn efficiently, as shown in our experiments (Section 6.2). In addition, compared to the analysis in Yang et al. (2020), Theorem 5.1 shows the existence of the step size for each iteration. **(3) Problem.** Finally, our work tackles a *different* problem compared to PCPO (which only ensures safety). We focus on how to safely and efficiently learn from an existing baseline policy, which is more conducive to practical applications of safe RL.



(a) Gather (b) Circle (c) Grid (d) Bottleneck (e) Car-racing (f) Demo.

Figure 2. **(a)** Gather: the agent is rewarded for gathering green apples, but is constrained to collect a limited number of red apples (Achiam et al., 2017). **(b)** Circle: the agent is rewarded for moving in a specified wide circle, but is constrained to stay within a safe region smaller than the radius of the circle (Achiam et al., 2017). **(c)** Grid: the agent controls the traffic lights in a grid road network and is rewarded for high throughput, but is constrained to let lights stay red for at most 7 consecutive seconds (Vinitsky et al., 2018). **(d)** Bottleneck: the agent controls a set of autonomous vehicles (shown in red) in a traffic merge situation and is rewarded for achieving high throughput, but constrained to ensure that human-driven vehicles (shown in white) have low speed for no more than 10 seconds (Vinitsky et al., 2018). **(e)** Car-racing: the agent controls an autonomous vehicle on a race track and is rewarded for driving through as many tiles as possible, but is constrained to use the brakes at most 5 times to encourage a smooth ride (Brockman et al., 2016). **(f)** A human player plays car-racing with demonstration data logged. These tasks are to show the applicability of our approach to a diverse set of problems.

6. Experiments

Our experiments study the following three questions: **(1)** How does SPACE perform compared to other baselines in behavior cloning and safe RL in terms of learning efficiency and constraint satisfaction? **(2)** How does SPACE trained with sub-optimal π_B perform (*e.g.,* human demonstration)? **(3)** How does the step 2 in SPACE affects the performance?

6.1. Setup

Tasks. We compare the proposed algorithm with existing approaches on five control tasks: three tasks with safety constraints ((a), (b) and (e) in Fig. 2), and two tasks with fairness constraints ((c) and (d) in Fig. 2). These tasks are briefly described in the caption of Fig. 2. We chose the traffic management tasks since a good control policy can benefit millions of drivers. In addition, we chose the car-racing task since a good algorithm should safely learn from baseline human policies. For all the algorithms, we use neural networks to represent Gaussian policies. We use the KL-divergence projection in the Mujoco and car-racing tasks, and the 2-norm projection in the traffic management task since it achieves better performance. We use a grid-search to select for the hyper-parameters. See the supplementary material for more experimental details.

Baseline policies π_B. To test whether SPACE can safely and efficiently leverage the baseline policy, we consider *three variants* of the baseline policies.

(1) *Sub-optimal* π_B^{cost} with $J_C(\pi_B^{\text{cost}}) \approx 0$.

(2) *Sub-optimal* π_B^{reward} with $J_C(\pi_B^{\text{reward}}) > h_C$.

(3) π_B^{near} with $J_C(\pi_B^{\text{near}}) \approx h_C$ (*i.e.,* the baseline policy has the same cost constraint as the agent, but is not guaranteed to have an optimal reward performance).

These π_B have *different* degrees of constraint satisfaction. This is to examine whether SPACE can safely learn from sub-optimal π_B. In addition, in the car-racing task we pre-train π_B using an off-policy algorithm (DDPG (Lillicrap et al., 2016)), which directly learns from *human demonstration data* (Fig. 2(f)). This is to demonstrate that π_B may come from a teacher or demonstration data. This *sub-optimal* human baseline policy is denoted by π_B^{human}.

For the ease of computation, we update h_D using $v \cdot (J_C(\pi^k) - h_C)^2 + h_D^k$ from Lemma 4.1, with a constant $v > 0$. We found that the performance is not heavily affected by v since we will update h_D at later iteration. The ablation studies of v can be found in Appendix E.1.

Baseline algorithms. Our goal is to study how to *safely* and *efficiently* learn from *sub-optimal* (possibly unsafe) baseline policies. We compare SPACE with five baseline methods that combine behavior cloning and safe RL algorithms.

(1) *Fixed-point Constrained Policy Optimization (f-CPO).* In f-CPO, we add the divergence objective in the reward function. The weight λ is fixed followed by a CPO update (optimize the reward and divergence cost w.r.t. the trust region and the cost constraints). The f-CPO policy update solves (Achiam et al., 2017):

$$\begin{aligned}\boldsymbol{\theta}^{k+1} = \arg\max_{\boldsymbol{\theta}} \; & (\boldsymbol{g}^k + \lambda \boldsymbol{a}^k)^T(\boldsymbol{\theta} - \boldsymbol{\theta}^k) \\ \text{s.t.} \; & \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^k)^T \boldsymbol{F}^k (\boldsymbol{\theta} - \boldsymbol{\theta}^k) \leq \delta \\ & {\boldsymbol{c}^k}^T(\boldsymbol{\theta} - \boldsymbol{\theta}^k) + d^k \leq 0.\end{aligned}$$

(2) *Fixed-point PCPO (f-PCPO).* In f-PCPO, we add the divergence objective in the reward function. The weight λ is fixed followed by a PCPO update (two-step process: optimize the reward and divergence cost, followed by the

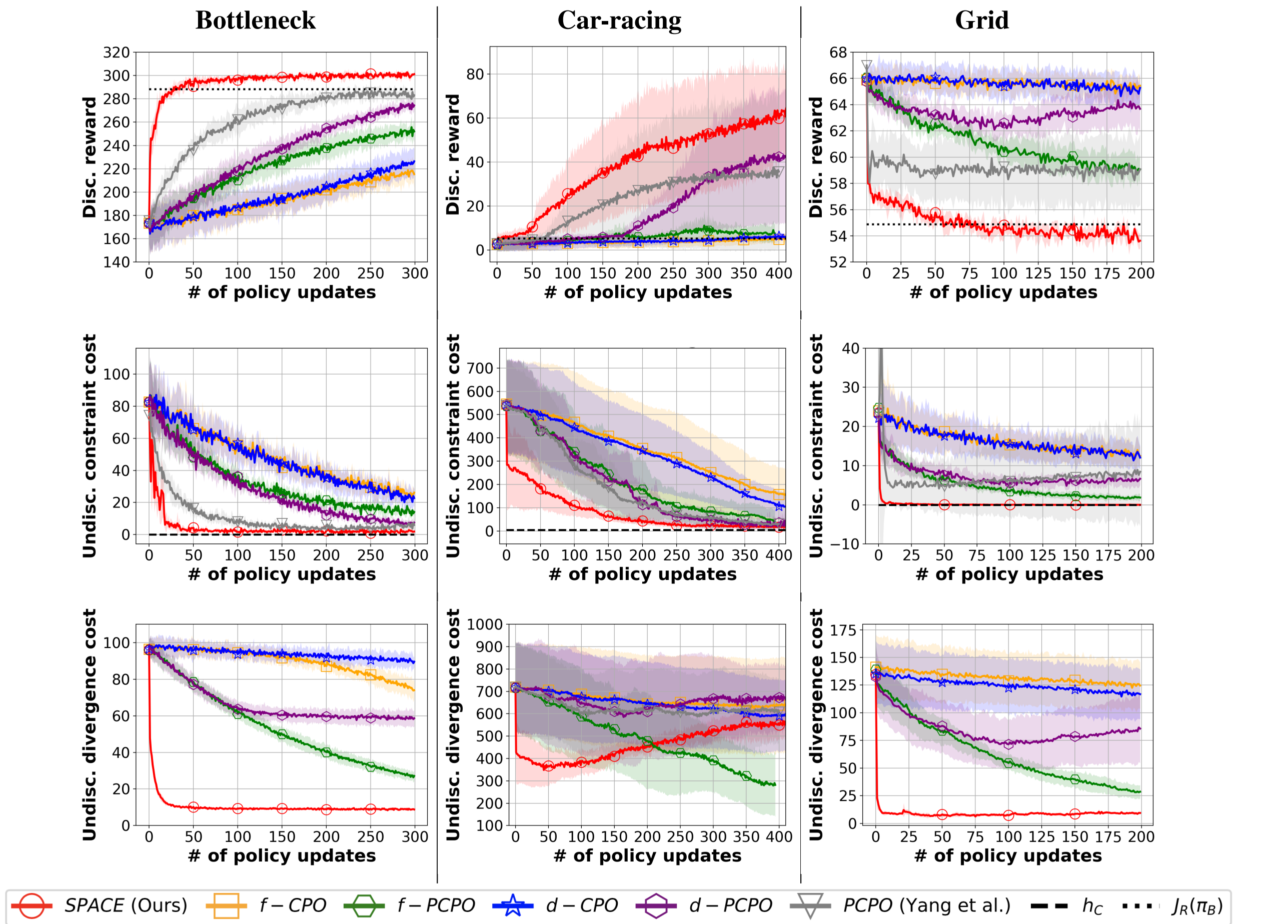


Figure 3. The discounted reward, the undiscounted constraint cost, and the undiscounted divergence cost over policy updates for the tested algorithms and tasks. The solid line is the mean and the shaded area is the standard deviation over 5 runs (random seed). The baseline policies in the grid and bottleneck tasks are π_B^{near}, and the baseline policy in the car-racing task is π_B^{human}. The black dashed line is the **cost constraint threshold** h_C. We observe that SPACE is the only algorithm that satisfies the constraints while achieving superior reward performance. Although π_B^{human} has substantially low reward, SPACE still can learn to improve the reward. (We show the results in these tasks as representative cases since they are more challenging. Please read Appendix for more results. Best viewed in color.)

projection to the safe set). The f-PCPO policy update solves:

$$
\begin{aligned}
\boldsymbol{\theta}^{k+\frac{1}{2}} &= \arg\max_{\boldsymbol{\theta}} \ (\boldsymbol{g}^k + \lambda \boldsymbol{a}^k)^T(\boldsymbol{\theta} - \boldsymbol{\theta}^k) \\
\text{s.t. } & \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^k)^T \boldsymbol{F}^k (\boldsymbol{\theta} - \boldsymbol{\theta}^k) \leq \delta, \quad \text{(trust region)} \\
\boldsymbol{\theta}^{k+1} &= \arg\min_{\boldsymbol{\theta}} \ \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^{k+\frac{1}{2}})^T \boldsymbol{L} (\boldsymbol{\theta} - \boldsymbol{\theta}^{k+\frac{1}{2}}) \\
\text{s.t. } & \boldsymbol{c}^{k^T}(\boldsymbol{\theta} - \boldsymbol{\theta}^k) + d^k \leq 0. \quad \text{(cost constraint)}
\end{aligned}
$$

(3) *Dynamic-point Constrained Policy Optimization (d-CPO).* The d-CPO update solves f-CPO problem with a stateful $\lambda^{k+1} = (\lambda^k)^\beta$, where $0 < \beta < 1$. This is inspired by Rajeswaran et al. (2017), in which they have the same weight-scheduling method to adjust λ^k.

(4) *Dynamic-point PCPO (d-PCPO).* The d-PCPO update solves f-PCPO problem with a stateful $\lambda^{k+1} = (\lambda^k)^\beta$, where $0 < \beta < 1$.

For all the experiments and the algorithms, the weight is fixed and it is set to 1. Note that both d-CPO and d-PCPO regularize the standard RL objective with the distance w.r.t. the baseline policy and make the regularization parameter (*i.e.,* λ) fade over time. This is a common practice to learn from the baseline policy. In addition, in many real applications you cannot have access to parameterized π_B (*e.g.,* neural network policies) or you want to design a policy with different architectures than π_B. Hence in our setting, we cannot directly initialize the learning policy with the baseline policy and then fine-tune it.

6.2. Results

Overall performance. The learning curves of the discounted reward ($J_R(\pi)$), the undiscounted constraint cost

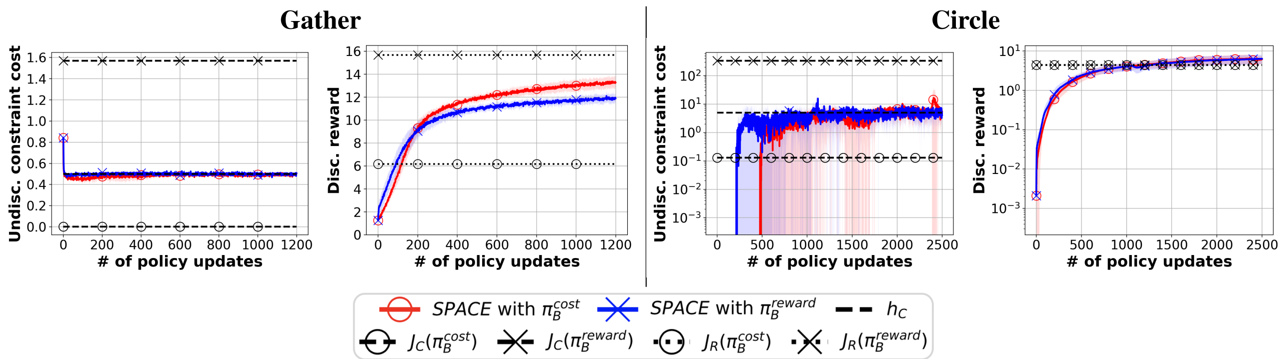


Figure 4. Learning from *sub-optimal* π_B. The undiscounted constraint cost and the discounted reward over policy updates for the gather and the circle tasks. The solid line is the mean and the shaded area is the standard deviation over 5 runs. The black dashed line is the cost constraint threshold h_C. We observe that SPACE satisfies the cost constraints even when learning from the sub-optimal π_B.

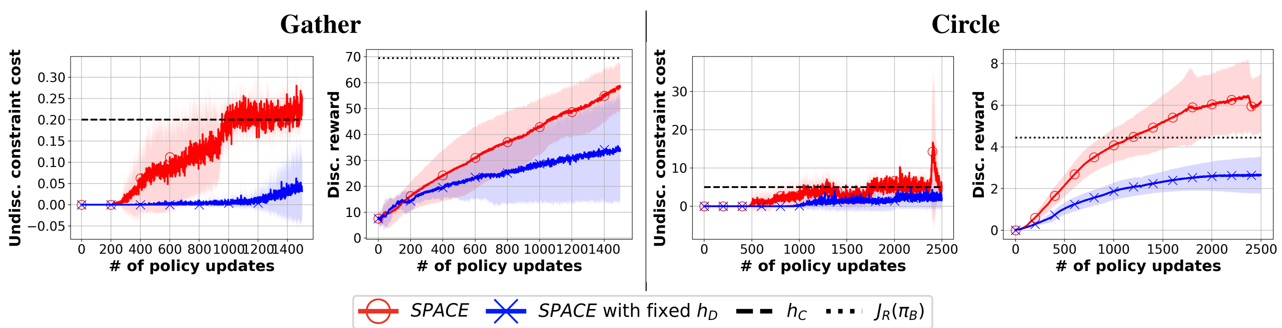


Figure 5. Ablation studies on the fixed h_D. The undiscounted constraint cost and the discounted reward over policy updates for the gather and the circle tasks. The solid line is the mean and the shaded area is the standard deviation over 5 runs. The black dashed line is the cost constraint threshold h_C. We observe that the update rule is critical for ensuring the learning performance improvement.

$(J_C(\pi))$, and the undiscounted divergence cost $(J_D(\pi))$ over policy updates are shown for all tested algorithms and tasks in Fig. 3. We use π_B^{near} in bottleneck and grid tasks, and π_B^{human} in car-racing task. Note that π_B^{human} from human demonstration is *highly sub-optimal* to the agent (*i.e.,* $J_R(\pi_B^{\text{human}})$ is small). The value of the reward is only around 5 as shown in the plot. It does not solve the task at hand. Overall, we observe that **(1)** SPACE achieves at least 2 times faster cost constraint satisfaction in all cases even learning from π_B^{human}. **(2)** SPACE achieves at least 10% more reward in the bottleneck and car-racing tasks compared to the best baseline, and **(3)** SPACE is the only algorithm that satisfies the cost constraints in all cases. In contrast, even if f(d)-CPO and f(d)-PCPO (similar to behavior cloning) are provided with good baseline policies π_B^{near}, they do not learn efficiently due to the conflicting reward and cost objectives. In addition, PCPO are less sample-efficient, which shows the accelerated learning of SPACE.

For example, in the car-racing task we observe that $J_D(\pi)$ in SPACE decreases at the initial iteration, but increases in the end. This implies that the learned policy is guided by the baseline policy π_B^{human} in the beginning, but use less supervision in the end. In addition, in the grid task we observe that the final reward of SPACE is lower than the baseline algorithm. This is because that SPACE converges to a policy in the cost constraint set, whereas the baseline algorithms do not find constraint-satisfying policies. Furthermore, we observe that $J_D(\pi)$ in the traffic tasks decreases throughout the training. This implies that SPACE intelligently adjusts h_D^k w.r.t. the performance of π_B to achieve safe learning.

f-CPO and f-PCPO. f-CPO and f-PCPO fail to improve the reward and have more cost violations. Most likely this is due to persistent supervision from the baseline policies which need not satisfy the cost constraints nor have high reward. For example, in the car-racing task we observe that the value of the divergence cost decreases throughout the training. This implies that the learned policy overly evolves to the sub-optimal π_B and hence degrades the reward performance.

d-CPO and d-PCPO. d-CPO and d-PCPO improve the reward slowly and have more cost violations. They do not use projection to quickly learn from π_B. For example, in the car-racing task $J_D(\pi)$ in d-CPO and d-PCPO are high compared to SPACE throughout the training. This suggests that simply regularizing the RL objective with the faded weight is susceptible to a sub-optimal π_B. In contrast to this heuristic, we use Lemma 4.1 to update h_D when needed, allowing π_B to influence the learning of the agent at any iterations depending on the learning progress of the agent.

Importantly, in our setup the agent does not have any prior knowledge about π_B. The agent has to stay close to π_B to verify its reward and cost performance. It is true that π_B may be constraint-violating, but it may also provide a useful signal for maximizing the reward. For example, in the grid task (Fig. 3), although π_B does not satisfy the cost constraint, it still helps the SPACE agent (by being close to π_B) to achieve faster cost satisfaction.

Having demonstrated the overall effectiveness of SPACE, our remaining experiments explore **(1)** SPACE's ability to safely learn from sub-optimal polices, and **(2)** the importance of the update method in Lemma 4.1. For compactness, we restrict our consideration on SPACE and the Mujoco tasks, which are widely used in RL community.

Sub-optimal π_B^{cost} and π_B^{reward}. Next, we test whether SPACE can learn from sub-optimal π_B. The learning curves of $J_C(\pi)$ and $J_R(\pi)$ over policy updates are shown for the gather and circle tasks in Fig. 4. We use two *sub-optimal* π_B: π_B^{cost} and π_B^{reward}, and learning agent's h_C is set to 0.5 (*i.e.,* π_B do not solve the task at hand). We observe that SPACE robustly satisfies the cost constraints in all cases even when learning from π_B^{reward}. In addition, we observe that learning guided by π_B^{reward} achieves faster reward learning efficiency at the *initial* iteration. This is because $J_R(\pi_B^{\text{reward}}) > J_R(\pi_B^{\text{cost}})$ as seen in the reward plot. Furthermore, we observe that learning guided by π_B^{cost} achieves faster reward learning efficiency at the *later* iteration. This is because by starting in the interior of the cost constraint set (*i.e.,* $J_C(\pi_B^{\text{cost}}) \approx 0 \leq h_C$), the agent can safely exploit the baseline policy. The results show SPACE enables fast convergence to a constraint-satisfying policy, even if π_B does not meet the constraint or does not optimize the reward.

SPACE with fixed h_D. In our final experiments, we investigate the importance of updating h_D when learning from a sub-optimal π_B. The learning curves of the $J_C(\pi)$ and $J_R(\pi)$ over policy updates are shown for the gather and circle tasks in Fig. 5. We observe that SPACE with fixed h_D converges to less reward. For example, in the circle task SPACE with the dynamic h_D achieves 2.3 times more reward. This shows that π_B in this task is highly sub-optimal to the agent and the need of using stateful h_D^k.

Moreover, Fig. 6 shows the divergence cost $J_D(\pi)$ and the value of h_D over the iterations in the car-racing task. We observe that SPACE gradually increases h_D to improve reward and cost satisfaction performance.

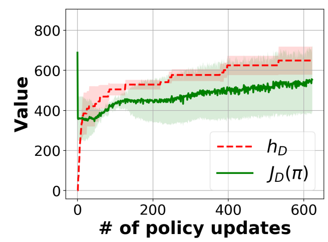


Figure 6. The divergence cost $J_D(\pi)$ and the value of h_D over the iterations in the car-racing task. We see that SPACE controls h_D to ensure divergence constraint satisfaction.

7. Conclusion

In this work, we addressed the problem of learning constraint-satisfying policies given potentially sub-optimal baseline policies. We explicitly analyzed how to safely learn from the baseline policy, and hence proposed an iterative policy optimization algorithm that alternates between maximizing expected return on the task, minimizing distance to the baseline policy, and projecting the policy onto the constraint-satisfying set. Our algorithm efficiently learns from a baseline policy as well as human provided demonstration data and achieves superior reward and cost performance compared with state-of-the-art approaches (*i.e.,* PCPO).

No algorithm is without limitations. Future work could improve SPACE in several ways. For instance, in Lemma 4.1, we do not guarantee that SPACE will increase h_D enough for the region around the baseline policy to contain the *optimal* policy. This is challenging since the optimization problem is non-convex. One possible solution is to rerun SPACE multiple times and reinitialize π_B with the previous learned policy each time. One evidence to support this method is that in the bottleneck task (Fig. 3), the agent trained with SPACE outperforms PCPO agent by achieving higher rewards and faster constraint satisfaction. The PCPO agent here can be seen as the SPACE agent trained without π_B. And then we train the SPACE agent with π_B from the learned PCPO agent. This shows that based on the learned policy, we can use SPACE to improve performance. In addition, it would be interesting to explore using other types of baseline policies such as rule-based policies and see how they impact the learning dynamics of SPACE.

Acknowledgements

The authors would like to thank members of the Princeton NLP Group, the anonymous reviewers, and the area chair for their comments. Tsung-Yen Yang thanks Siemens Corporation, Corporate Technology for their support.

References

Abbeel, P., Coates, A., and Ng, A. Y. Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research*, 29(13):1608–1639, 2010.

Abel, D., Salvatier, J., Stuhlmüller, A., and Evans, O. Agent-agnostic human-in-the-loop reinforcement learning. *arXiv preprint arXiv:1701.04079*, 2017.

Achiam, J., Held, D., Tamar, A., and Abbeel, P. Constrained policy optimization. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 22–31. PMLR, 2017.

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

Avni, G., Bloem, R., Chatterjee, K., Henzinger, T. A., Könighofer, B., and Pranger, S. Run-time optimization for learned controllers through quantitative games. In *International Conference on Computer Aided Verification*, pp. 630–649. Springer, 2019.

Balakrishna, A., Thananjeyan, B., Lee, J., Zahed, A., Li, F., Gonzalez, J. E., and Goldberg, K. On-policy robot imitation learning from a converging supervisor. In *Proceedings of the Conference on Robot Learning*, 2019.

Bartocci, E., Grosu, R., Katsaros, P., Ramakrishnan, C., and Smolka, S. A. Model repair for probabilistic systems. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pp. 326–340. Springer, 2011.

Bharadhwaj, H., Kumar, A., Rhinehart, N., Levine, S., Shkurti, F., and Garg, A. Conservative safety critics for exploration. *arXiv preprint arXiv:2010.14497*, 2020.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Brown, D. S., Goo, W., Nagarajan, P., and Niekum, S. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 783–792. PMLR, 2019.

Chen, G. and Teboulle, M. Convergence analysis of a proximal-like minimization algorithm using bregman functions. *SIAM Journal on Optimization*, 3(3):538–543, 1993.

Chernova, S. and Thomaz, A. L. Robot learning from human teachers. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 8(3):1–121, 2014.

Chow, Y., Nachum, O., Duéñez-Guzmán, E. A., and Ghavamzadeh, M. A lyapunov-based approach to safe reinforcement learning. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 8103–8112, 2018.

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

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

Driessens, K. and Džeroski, S. Integrating guidance into relational reinforcement learning. *Machine Learning*, 57 (3):271–304, 2004.

Duan, Y., Chen, X., Houthooft, R., Schulman, J., and Abbeel, P. Benchmarking deep reinforcement learning for continuous control. In Balcan, M. and Weinberger, K. Q. (eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 1329–1338. JMLR.org, 2016.

Fu, M. and Prashanth L, A. Risk-sensitive reinforcement learning: A constrained optimization viewpoint. *arXiv preprint arXiv:1810.09126*, 2018.

Fujimoto, S., Conti, E., Ghavamzadeh, M., and Pineau, J. Benchmarking batch deep reinforcement learning algorithms. *arXiv preprint arXiv:1910.01708*, 2019.

Gao, Y., Lin, J., Yu, F., Levine, S., and Darrell, T. Reinforcement learning from imperfect demonstrations. In *Proceedings of the International Conference on Machine Learning*, 2018.

Garcia, J. and Fernández, F. Safe exploration of state and action spaces in reinforcement learning. *Journal of Artificial Intelligence Research*, 45:515–564, 2012.

Garcia, J. and Fernandez, F. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 2672–2680, 2014.

Hasanbeig, M., Abate, A., and Kroening, D. Cautious reinforcement learning with logical constraints. *arXiv preprint arXiv:2002.12156*, 2020.

Hester, T., Vecerík, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., Horgan, D., Quan, J., Sendonaris, A., Osband, I., Dulac-Arnold, G., Agapiou, J. P., Leibo, J. Z., and Gruslys, A. Deep q-learning from demonstrations. In McIlraith, S. A. and Weinberger, K. Q. (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 3223–3230. AAAI Press, 2018.

Ho, J. and Ermon, S. Generative adversarial imitation learning. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 4565–4573, 2016.

Jansen, N., Könighofer, B., Junges, S., Serban, A., and Bloem, R. Safe Reinforcement Learning Using Probabilistic Shields (Invited Paper). In Konnov, I. and Kovács, L. (eds.), *31st International Conference on Concurrency Theory (CONCUR 2020)*, volume 171 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 3:1–3:16, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. ISBN 978-3-95977-160-3. doi: 10.4230/LIPIcs.CONCUR.2020.3.

Jaques, N., Ghandeharioun, A., Shen, J. H., Ferguson, C., Lapedriza, A., Jones, N., Gu, S., and Picard, R. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*, 2019.

Junges, S., Jansen, N., Dehnert, C., Topcu, U., and Katoen, J.-P. Safety-constrained reinforcement learning for mdps. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pp. 130–146. Springer, 2016.

Kakade, S. M. and Langford, J. Approximately optimal approximate reinforcement learning. In Sammut, C. and Hoffmann, A. G. (eds.), *Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002), University of New South Wales, Sydney, Australia, July 8-12, 2002*, pp. 267–274. Morgan Kaufmann, 2002.

Koppejan, R. and Whiteson, S. Neuroevolutionary reinforcement learning for generalized control of simulated helicopters. *Evolutionary intelligence*, 4(4):219–241, 2011.

Kwon, M., Biyik, E., Talati, A., Bhasin, K., Losey, D. P., and Sadigh, D. When humans aren't optimal: Robots that collaborate with risk-aware humans. In *Proceedings of ACM/IEEE International Conference on Human-Robot Interaction*, 2020.

Le, H. M., Voloshin, C., and Yue, Y. Batch policy learning under constraints. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3703–3712. PMLR, 2019.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. In Bengio, Y. and LeCun, Y. (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

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

Mokhtari, A., Ozdaglar, A. E., and Jadbabaie, A. Escaping saddle points in constrained optimization. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 3633–3643, 2018.

Mülling, K., Kober, J., Kroemer, O., and Peters, J. Learning to select and generalize striking movements in robot table tennis. *The International Journal of Robotics Research*, 32(3):263–279, 2013.

Pathak, S., Ábrahám, E., Jansen, N., Tacchella, A., and Katoen, J.-P. A greedy approach for the efficient repair of stochastic models. In *NASA Formal Methods Symposium*, pp. 295–309. Springer, 2015.

Quintía Vidal, P., Iglesias Rodríguez, R., Rodríguez González, M. Á., and Vázquez Regueiro, C. Learning on real robots from experience and simple user feedback. 2013.

Rajeswaran, A., Kumar, V., Gupta, A., Vezzani, G., Schulman, J., Todorov, E., and Levine, S. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Proceedings of Robotics: Science and Systems*, 2017.

Ross, S., Gordon, G., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 627–635, 2011.

Schulman, J., Levine, S., Abbeel, P., Jordan, M. I., and Moritz, P. Trust region policy optimization. In Bach, F. R. and Blei, D. M. (eds.), *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 1889–1897. JMLR.org, 2015.

Schulman, J., Moritz, P., Levine, S., Jordan, M. I., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. In Bengio, Y. and LeCun, Y. (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484, 2016.

Smart, W. D. and Kaelbling, L. P. Practical reinforcement learning in continuous spaces. In Langley, P. (ed.), *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pp. 903–910. Morgan Kaufmann, 2000.

Srinivasan, K., Eysenbach, B., Ha, S., Tan, J., and Finn, C. Learning to be safe: Deep rl with a safety critic. *arXiv preprint arXiv:2010.14603*, 2020.

Stooke, A., Achiam, J., and Abbeel, P. Responsive safety in reinforcement learning by PID lagrangian methods. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9133–9143. PMLR, 2020.

Sui, Y., Gotovos, A., Burdick, J. W., and Krause, A. Safe exploration for optimization with gaussian processes. In Bach, F. R. and Blei, D. M. (eds.), *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 997–1005. JMLR.org, 2015.

Sun, W., Gordon, G. J., Boots, B., and Bagnell, J. A. Dual policy iteration. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 7059–7069, 2018.

Tessler, C., Mankowitz, D. J., and Mannor, S. Reward constrained policy optimization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Thananjeyan, B., Balakrishna, A., Rosolia, U., Li, F., McAllister, R., Gonzalez, J. E., Levine, S., Borrelli, F., and Goldberg, K. 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.

Thananjeyan, B., Balakrishna, A., Nair, S., Luo, M., Srinivasan, K., Hwang, M., Gonzalez, J. E., Ibarz, J., Finn, C., and Goldberg, K. Recovery rl: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6(3):4915–4922, 2021.

Turchetta, M., Berkenkamp, F., and Krause, A. Safe exploration in finite markov decision processes with gaussian processes. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 4305–4313, 2016.

Vecerik, M., Hester, T., Scholz, J., Wang, F., Pietquin, O., Piot, B., Heess, N., Rothörl, T., Lampe, T., and Riedmiller, M. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817*, 2017.

Vinitsky, E., Kreidieh, A., Le Flem, L., Kheterpal, N., Jang, K., Wu, C., Wu, F., Liaw, R., Liang, E., and Bayen, A. M. Benchmarks for reinforcement learning in mixed-autonomy traffic. In *Proceedings of Conference on Robot Learning*, pp. 399–409, 2018.

Wachi, A. and Sui, Y. Safe reinforcement learning in constrained markov decision processes. In *Proceedings of* *the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9797–9806. PMLR, 2020.

Wu, Y., Charoenphakdee, N., Bao, H., Tangkaratt, V., and Sugiyama, M. Imitation learning from imperfect demonstration. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6818–6827. PMLR, 2019.

Yang, T., Rosca, J., Narasimhan, K., and Ramadge, P. J. Projection-based constrained policy optimization. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Zhang, J., Cheung, B., Finn, C., Levine, S., and Jayaraman, D. Cautious adaptation for reinforcement learning in safety-critical settings. In *International Conference on Machine Learning*, pp. 11055–11065. PMLR, 2020.

Zhang, R., Torabi, F., Guan, L., Ballard, D. H., and Stone, P. Leveraging human guidance for deep reinforcement learning tasks. In Kraus, S. (ed.), *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pp. 6339–6346. ijcai.org, 2019. doi: 10.24963/ijcai.2019/884.

Zheng, L. and Ratliff, L. J. Constrained upper confidence reinforcement learning. *arXiv preprint arXiv:2001.09377*, 2020.