2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Safety-Constrained Policy Transfer with Successor Features

Zeyu Feng[1], Bowen Zhang[1], Jianxin Bi[1], Harold Soh[1,2]

***Abstract*— In this work, we focus on the problem of safe policy transfer in reinforcement learning: we seek to leverage existing policies when learning a new task with specified constraints. This problem is important for safety-critical applications where interactions are costly and unconstrained exploration can lead to undesirable or dangerous outcomes, e.g., with physical robots that interact with humans. We propose a Constrained Markov Decision Process (CMDP) formulation that simultaneously enables the transfer of policies and adherence to safety constraints. Our formulation cleanly separates task goals from safety considerations and permits the specification of a wide variety of constraints. Our approach relies on a novel extension of generalized policy improvement to constrained settings via a Lagrangian formulation. We devise a dual optimization algorithm that estimates the optimal dual variable of a target task, thus enabling safe transfer of policies derived from successor features learned on source tasks. Our experiments in simulated domains show that our approach is effective; it visits unsafe states less frequently and outperforms alternative state-of-the-art methods when taking safety constraints into account.**

I. Introduction

Reinforcement learning (RL) is an attractive paradigm for developing agents that learn through interactions with an environment—the framework is not only elegant and principled, but also supported by recent empirical evidence showing RL agents can achieve superhuman performance on complex tasks [1]. Nevertheless, RL remains challenging to employ in environments (especially real-world settings) where interactions are costly and safety is a principal concern. The cause for this can be traced back to two well-known deficiencies of canonical RL methods: the high sample complexity associated with learning and the risks associated with unconstrained exploration. Recognition of this problem has given rise to a body of work on Safe-RL [2], which attempt to maximize expected returns while ensuring thresholds on system performance and safety are met.

In this paper, we build upon this line of work and investigate how *safety-constrained* policies can be *transferred* across tasks with different objectives. Specifically, we develop transfer learning for Constrained Markov Decision Processes (CMDPs) [3]. Our approach is motivated by the advantages afforded by the constrained criterion compared to other safety formulations: CMDPs are well-studied models that cleanly separate task goals from safety considerations. Similar to very recent work on risk-aware policy transfer [4], we employ the powerful successor features (SF) [5] framework as our main mechanism for transfer. But unlike [4], our setup naturally encompasses cases where risk cannot be captured by the variance of the return and enables the specification of a wide variety of safety considerations. In essence, transfer in CMDPs promotes safe behavior via both sample reduction and constraint adherence.

Our safe-policy transfer framework rests upon a Lagrangian formulation of CMDP with successor features. In contrast to the unconstrained and risk-aware settings, our error analysis reveals the importance of the dual variable to induce safe policy transfer. With this insight, we introduce a dual optimization method for approximating the optimal Lagrangian multiplier, along with a proof of convergence. Taken together, our theoretical findings and algorithmic development provides a principled approach for safe-policy transfer.

Our framework—which we call the **S**uccessor **F**eature **T**ransfer for **Co**nstrained **P**olicies (SFT-CoP)—enables agents to exploit structure in both the task objectives and constraints to more quickly learn safe and optimal behavior on new tasks. Experiments on three benchmark problems [4], [5]—a discrete gridworld environment and two continuous physical robot simulations—show SFT-CoP is able to achieve positive transfer in terms of both task objectives and safety considerations. In particular, we see that SFT-CoP compares favorably to safety-agnostic transfer and a state-of-the-art risk-sensitive method. In summary, this paper makes three key contributions:

- A generalized policy improvement theorem for constrained policy transfer via successor features;
- A dual optimization algorithm (with a proof of convergence) that enables safe transfer of source policies to a new target task;
- Experimental results on benchmark RL problems that validate the effectiveness of transfer learning with safety constraints.

II. Preliminaries

In this section, we give a succinct review of background material, specifically Markov Decision Processes (MDPs) and Constrained MDPs (CMDPs). For more information, please refer to excellent survey articles [6], [3].

Markov Decision Process (MDP) We begin with the popular MDP framework [6] for modeling sequential decision-making problems. Formally, an MDP is denoted as a tuple $M = (\mathcal{S}, \mathcal{A}, p, r, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is an action set, p is a transition probability function, r is a reward function and γ is a discount factor. After an agent executes an action a in state s, it transitions to a new state s' and receives a reward. This transition is governed by

[1]Dept. of Computer Science, National University of Singapore {zeyu, bowenzhang, jianxin.bi, harold}@comp.nus.edu.sg.
[2]Smart Systems Institute (SSI), NUS.

$p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, \infty]$, which models the distribution of the next state $p(s'|s,a)$. The function $r(s,a,s')$ provides the reward, which we assume to be bounded. An agent's policy $\pi \in \Pi$ dictates its action given a particular state and we focus on stochastic policies where $\pi(s) = p(a|s)$.

Given an MDP, we seek the optimal policy π^* that maximizes the expected discounted sum of rewards or value function $\max_{\pi \in \Pi} V_r^\pi(s_0) = \mathbb{E}_\pi[R_r|s_0]$, where $R_r = \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1})$ is the reward return for a specific trajectory. The expectation above is taken with respect to trajectories generated by applying the policy π in the environment starting from an initial state s_0. To ease our exposition, we assume the initial state to be fixed, but our work readily extends to the more general case where s_0 is drawn from a prior distribution of starting states. The value function V_r^π is related to the *action-value* function Q_r^π, where $V_r^\pi(s_0) = \mathbb{E}_{a \sim \pi(s_0)} Q_r^\pi(s_0, a)$ and $Q_r^\pi(s_0, a) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1}) | s_0, a_0 = a]$.

Given the action-value function Q_r^π of a particular policy π, Bellman's celebrated *policy improvement theorem* [7] provides a means of obtaining a policy π' that is guaranteed to be no worse than π and possibly better. The policy π' is obtained by acting greedily with respect to Q_r^π, *i.e.*, $\pi'(s) \in \operatorname{argmax}_a Q_r^\pi(s,a)$. Under certain conditions, successive evaluation of Q_r^π and improvement leads to an optimal policy π^* — this forms the basis of RL methods that employ dynamic programming.

Constrained MDP (CMDP) In many decision-making scenarios, constraints play an important role, *e.g.*, to account for safety or ethical considerations. To model such problems, we can extend the MDP setup above to include a constraint return $R_c = \sum_{t=0}^{\infty} \gamma^t c(s_t, a_t, s_{t+1})$ of some utility function values $c(s,a,s') \in [0,1]$. For example, the utility function c can be an indicator function $c(s,a,s') = \mathbb{1}(s \in S)$ of whether an agent is in a safe set S [8].

The CMDP $M^c = (\mathcal{S}, \mathcal{A}, p, r, c, \gamma)$ [3], has corresponding value $V_c^\pi(s_0) = \mathbb{E}_{a \sim \pi(s_0)}[Q_c^\pi(s_0, a)]$ and action-value functions $Q_c^\pi(s_0, a) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t, s_{t+1}) | s_0, a_0 = a]$ associated with the constraint function c. Our goal is to obtain an optimal policy that maximizes the expected reward return while ensuring the expected *utility return* is met,

$$\max_{\pi \in \Pi} \quad V_r^\pi(s_0) \qquad \text{s.t.} \quad V_c^\pi(s_0) \geq \tau. \tag{1}$$

Here, τ is the threshold we seek to enforce for the constraint[1]. Prior work has shown that constraining the utility value function as above is a reasonable approach for safety [9]; returning to our indicator function example above, setting $V_c^\pi(s_0) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t \mathbb{1}(s_t \in S) | s_0]$ provides a guarantee on the probability of the agent being in the safe set S when following the policy π [8]. Note that we can easily accommodate (i) *cost* constraints $\tilde{c}$ via a simple transformation to utilities $c(s_t, a_t, s_{t+1}) = 1 - \tilde{c}(s_t, a_t, s_{t+1})$ and (ii) multiple constraints by specifying a set of thresholds $\{\tau^k\}_{k=1}^K$ for K expected utility returns $\{V_{c^k}^\pi(s_0)\}_{k=1}^K$.

[1]We set $\tau \in (0, 1/1-\gamma]$ to avoid trivial solutions.

III. Policy transfer for CMDPs

In this work, we aim to transfer a set of policies, specifically Q functions learned on a set of source CMDP tasks, to a new target task. In the RL setting, the agent only observes scalar rewards/utilities when interacting with the environment, i.e., the agent does not know the reward and constraint functions and has to quickly learn them.

Our approach is based on the successor features (SFs) framework [5] where the source and target tasks occur within the same environment (with the same transition function), but have different reward functions. Applied to the standard MDP framework, the set of tasks can be expressed as

$$\begin{aligned} &\mathcal{M}_{\boldsymbol{\phi}}(\mathcal{S}, \mathcal{A}, p, \gamma) = \\ &\quad \{M_i(\mathcal{S}, \mathcal{A}, p, r_i, \gamma) | r_i(s,a,s') = \boldsymbol{\phi}(s,a,s')^\top \mathbf{w}_{r,i}\}, \end{aligned} \tag{2}$$

where each task $M_i \in \mathcal{M}_{\boldsymbol{\phi}}$ is associated with a specific reward function r_i [10]. A key assumption is that the reward r_i can be computed using a shared feature function $\boldsymbol{\phi}(s,a,s') \in \mathbb{R}^d$ and a unique weight vector $\mathbf{w}_{r,i} \in \mathbb{R}^d$ [5].

The advantage of using the model (2) is that the action-value function is given by $Q_{r,i}^{\pi_i}(s,a) = \boldsymbol{\psi}^{\pi_i}(s,a)^\top \mathbf{w}_{r,i}$ where $\boldsymbol{\psi}^{\pi_i}$ are the *successor features*. In other words, we can quickly evaluate a policy π_i on a new task M_j by computing a dot-product. The successor features $\boldsymbol{\psi}^{\pi_i}(s,a)$ satisfy the Bellman equation and can be computed via dynamic programming, and $\mathbf{w}_{r,j}$ can be directly plugged-in (if known) or learned using standard learning algorithms. This property is especially desirable for transfer learning where we are given N source tasks $\{M_i\}_{i=1}^N$ from $\mathcal{M}_{\boldsymbol{\phi}}$ with their corresponding action-value functions $\{Q_{r,i}^{\pi_i}\}_{i=1}^N$, and seek to rapidly learn a new task $M_{N+1} \in \mathcal{M}_{\boldsymbol{\phi}}$. Under certain conditions, the generalized policy improvement (GPI) theorem guarantees that the policy $\pi(s) \in \operatorname{argmax}_a \max_i Q_{r,j}^{\pi_i}(s,a)$ is no worse than the individual source policies.

To account for safety, we extend the above SF framework to CMDPs, where the set of tasks,

$$\begin{aligned} &\mathcal{M}_{\boldsymbol{\phi}}^c(\mathcal{S}, \mathcal{A}, p, \gamma) = \{M^c(\mathcal{S}, \mathcal{A}, p, r_i, c_i, \gamma) | r_i(s,a,s') \\ &\quad = \boldsymbol{\phi}(s,a,s')^\top \mathbf{w}_{r,i}, c_i(s,a,s') = \boldsymbol{\phi}(s,a,s')^\top \mathbf{w}_{c,i}\}, \end{aligned} \tag{3}$$

take on an additional utility weight vector $\mathbf{w}_{c,i}$ for each task. Without loss of generality, we use the same $\boldsymbol{\phi}$ function for both r and c[2]. We focus on a single constraint but the extension to multiple constraints is straightforward. We highlight that unlike the standard MDP setting, the optimal policy for a CMDP depends on *both* the reward function and constraints; a naive application of GPI does not guarantee that the transferred policy will adhere to the specified constraints. In our setup, we model the value functions of both the reward and utility functions using SFs, which enables efficient policy and constraint evaluation during transfer.

A. General policy improvement in CMDP

In the following analysis, we seek to better understand the conditions under which positive transfer can occur in

[2]We can handle the more general case of different $\boldsymbol{\phi}$ functions by concatenating their outputs into a single vector and expanding the corresponding weights, which conforms to Eq. (3).

CMDPs. We begin by revisiting the optimal policy π_j^* of a target CMDP M_j^c. The optimal value function is $V_{r,j}^{\pi_j^*}(s) = \max_{\pi\in\Pi}\left\{V_{r,j}^{\pi}(s)\,|V_{c,j}^{\pi}(s)\geq\tau\right\}$, where we see that the class of policies is restricted to those whose expected utility returns satisfy the threshold constraints. The dual function is

$$d(\lambda_j) = \max_{\pi\in\Pi} L(\pi,\lambda_j), \tag{4}$$

where $L(\pi,\lambda_j) = V_{r,j}^{\pi}(s) + \lambda_j\left(V_{c,j}^{\pi}(s)-\tau\right)$ is the Lagrangian and $\lambda_j \in \mathbb{R}_+$ is the Lagrange multiplier (or dual variable). The optimal dual variable minimizes the Lagrangian: $\lambda_j^* = \operatorname{argmin}_{\lambda_j\geq 0} d(\lambda_j)$, and as we will see, plays an important role during transfer.

Given the above, we extend GPI to the multi-task CMDP setting given in Eq. (3). In the following, we adopt the standard assumption that a feasible policy exists, *i.e.*,

Assumption 1. (Slater's condition). There exists a policy $\bar{\pi}\in\Pi$ such that $V_{c,j}^{\bar{\pi}}(s)\geq\tau$,

and introduce our first key result:

Proposition 1. *Let $Q_{r,j}^{\pi_i^*}$ and $Q_{c,j}^{\pi_i^*}$ be the action-value functions of an optimal policy of $M_i^c \in \mathcal{M}_{\boldsymbol{\phi}}^c$ when executed in $M_j^c \in \mathcal{M}_{\boldsymbol{\phi}}^c$. Given approximations $\left\{\tilde{Q}_{r,j}^{\pi_i}\right\}_{i=1}^N$ and $\left\{\tilde{Q}_{c,j}^{\pi_i}\right\}_{i=1}^N$ such that $\left|Q_{r,j}^{\pi_i^*}(s,a)-\tilde{Q}_{r,j}^{\pi_i}(s,a)\right|\leq\epsilon$ and $\left|Q_{c,j}^{\pi_i^*}(s,a)-\tilde{Q}_{c,j}^{\pi_i}(s,a)\right|\leq\epsilon$, $\forall s,a$ and $i\in\{1,\ldots,N\}$. Consider the policy*

$$\pi(s)\in\operatorname*{argmax}_a \max_{i\in[N]}\left\{\tilde{Q}_{r,j}^{\pi_i}(s,a)+\tilde{\lambda}_j\tilde{Q}_{c,j}^{\pi_i}(s,a)\right\}, \tag{5}$$

by specifying a $\tilde{\lambda}_j$ for the new task M_j^c. If Slater's condition (1) holds, we have

$$\begin{aligned}
&Q_{j,\lambda_j^*}^{\pi_j^*}(s,a) - Q_{j,\tilde{\lambda}_j}^{\pi}(s,a)\\
\leq&\min_i \frac{2}{1-\gamma}\Big(\boldsymbol{\phi}_{\max}\|\mathbf{w}_{r,j}-\mathbf{w}_{r,i}\| + \boldsymbol{\phi}_{\max}\|\lambda_j^*\mathbf{w}_{c,j}-\lambda_i^*\mathbf{w}_{c,i}\|\\
&+\left|\lambda_j^*-\tilde{\lambda}_j\right| + \epsilon(1+\tilde{\lambda}_j)\Big) + 2\tau\left|\lambda_j^*-\lambda_i^*\right|,
\end{aligned} \tag{6}$$

where $Q_{j,\lambda}^{\pi}(s,a) \coloneqq Q_{j,r}^{\pi}(s,a)+\lambda\left(Q_{j,c}^{\pi}(s,a)-\tau\right)$ for any π, j and λ, and $\boldsymbol{\phi}_{\max} = \max_{s,a}\|\boldsymbol{\phi}(s,a)\|$.

The detailed proofs along with additional experiments can be found in the supplementary materials[3]. The proposition above upper-bounds the differences in action-values of the transfer policy in Eq. (5) and the optimal policies of the target task M_j^c. The bound formally captures the roles of multiple factors in safe *transfer*. Similar to the GPI theorem with successor features, the first term $\|\mathbf{w}_{r,j}-\mathbf{w}_{r,i}\|$ characterizes the difference in the task objectives. However, our constrained version in Prop. 1 reveals the critical role of the optimal dual variable λ_j^* and its estimation $\tilde{\lambda}_j$ in bounding the loss. We draw attention to the remaining terms in the bound; we see that the second and last elements characterize the similarity between tasks in terms of their utilities and optimal dual variables. This reflects that intuition that the performance of the agent depends on the similarity to source tasks in terms of the safety constraints. The term $\left|\lambda_j^*-\tilde{\lambda}_j\right|$ highlights the gap caused by estimating the optimal dual variable; the larger the estimation error, the larger the loss incurred by the agent.

To shed additional light on the transfer policy, let us draw a connection to the optimal CMDP value function. We leverage the following lemma on strong duality[4],

Lemma 1. [Strong duality] [11]. *If Slater's condition (1) holds, then strong duality holds $V_{r,j}^{\pi_j^*}(s) = d\left(\lambda_j^*\right)$.*

Combining the above with the definition of the dual function in Eq. (4) yields

$$\begin{aligned}
V_{r,j}^{\pi_j^*}(s) &= \max_{\pi\in\Pi} V_{r,j}^{\pi}(s) + \lambda_j^*\left(V_{c,j}^{\pi}(s)-\tau\right)\\
&= \max_{\pi\in\Pi, a=\pi(s)} Q_{r,j}^{\pi}(s,a) + \lambda_j^*\left(Q_{c,j}^{\pi}(s,a)-\tau\right)
\end{aligned} \tag{7}$$

The second equality above holds because the optimal policies have corresponding optimal action-value functions, since the Lagrangian can be equivalently written as a value function $L\left(\pi,\lambda_j^*\right) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t r_j^{\lambda_j^*}(s_t,a_t,s_{t+1})\,|s_0=s\right]$ with a single reward function $r_j^{\lambda}(s,a,s') = r_j(s,a,s') + \lambda\left(c_j(s,a,s')-(1-\gamma)\tau\right)$. Let us compare Eq. (7) against the policy in Eq. (5). We see that the value of the policy matches that of the optimal value function if (i) $\tilde{\lambda}_j = \lambda_j^*$, (ii) the true action-value functions of source policies are available, and (iii) the set of source policies $\{\pi_i^*\}_{i=1}^N$ equals policy space Π (or at least covers the optimal region). While less formal than Prop. 1, this analysis reveals the ingredients needed for strong positive transfer (and potential sources of policy under-performance), which again includes the optimal dual λ_j^*.

Taken together, both Prop. 1 and the above discussion indicate that a good estimate of the optimal dual variable is important for transfer in CMDPs. We could learn it by training a policy on the target task but this would typically entail many interactions with the environment, defeating the purpose of transfer. Instead, we propose a practical optimization method to approximate λ_j^* *without* having to resort to RL on the target task.

B. Optimal dual estimation

Recall that the optimal dual variable for target M_j^c is:

$$\lambda_j^* \in \operatorname*{argmin}_{\lambda_j\geq 0}\max_{\pi\in\Pi} V_{r,j}^{\pi}(s) + \lambda_j\left(V_{c,j}^{\pi}(s)-\tau\right). \tag{8}$$

Given the source tasks, we can approximate the optimal dual via maximizing among source policies,

$$\hat{\lambda}_j \in \operatorname*{argmin}_{\lambda_j\geq 0}\max_{i\in[N]} V_{r,j}^{\tilde{\pi}_i}(s) + \lambda_j\left(V_{c,j}^{\tilde{\pi}_i}(s)-\tau\right), \tag{9}$$

where $\tilde{\pi}_i$ is the source policy given by the source Q functions ($\tilde{Q}_{r,i}^{\pi_i}$ and $\tilde{Q}_{c,i}^{\pi_i}$) and dual variable $\tilde{\lambda}_i$. However, strong duality does not hold for Eq. (9) because the domain of the primal problem is non-convex. As a workaround, we relax the domain of the primal problem (*i.e.*, the policy space) to allow stochastic combination of policies:

[3]Supplementary materials: `https://clear-nus.github.io/papers/sft_cop_appendix.pdf`

[4]This lemma is also used in the proof of Proposition 1.

Algorithm 1 Dual variable update for SFs transfer

Require: The number of iterations T, A sequence of step size $\eta^{(t)}$.

Input: Value functions of a set of source policies evaluated on M_j^c: $\left\{\hat{V}_{r,j}^{\tilde{\pi}_i}(s), \hat{V}_{c,j}^{\tilde{\pi}_i}(s)\right\}_{i=1}^{N}$.

Output: Approximate optimal dual variable $\lambda^{(T)}$.

1: **Initialization** $\lambda^{(1)} = 0$, $t = 1$
2: **while** $t \leq T$ **do**
3: $\quad i^{(t+1)} = \text{argmax}_{i\in[N]}\, \hat{V}_{r,j}^{\tilde{\pi}_i}(s) + \lambda^{(t)}\left(\hat{V}_{c,j}^{\tilde{\pi}_i}(s) - \tau\right)$
4: $\quad \lambda^{(t+1)} = \lambda^{(t)} - \eta^{(t)}\left(\hat{V}_{r,j}^{\tilde{\pi}_{i^{(t+1)}}}(s) - \tau\right)$
5: $\quad \lambda^{(t+1)} = \mathrm{P}_{\mathbb{R}_{\geq 0}}\left(\lambda^{(t+1)}\right)$
6: $\quad t \leftarrow t + 1$
7: **end while**

$\Pi_c := \texttt{Conv}\left(\{\tilde{\pi}_i\}_{i=1}^{N}\right)$. $\pi_\alpha(a|s) = \frac{\sum_{i=1}^{n}\alpha_i\rho^{\tilde{\pi}_i}(s,a)}{\int_{\mathcal{A}}\sum_{i=1}^{n}\alpha_i\rho^{\tilde{\pi}_i}(s,a)\,\mathrm{d}a} \in \Pi_c$ has value function $V_{r,j}^{\pi_\alpha}(s) = \sum_{i=1}^{N}\alpha_i V_{r,j}^{\tilde{\pi}_i}(s) = \sum_{i=1}^{n}\alpha_i \int_{\mathcal{S},\mathcal{A}} \rho^{\tilde{\pi}_i}(s,a) r_j(s,a)\,\mathrm{d}s\,\mathrm{d}a$, where $\sum_{i=1}^{N}\alpha_i = 1$, $\alpha_i \geq 0$ for any i and $\rho^\pi(s,a) = \sum_{t=0}^{\infty}\gamma^t P(s_t = s, a_t = a|\pi)$ is occupation measure. Effectively, we expand the space of the finite number of policies from source tasks and assume the existence of a feasible policy in this space,

Assumption 2. (Slater's condition for policy transfer). There exists a policy $\bar{\pi} \in \Pi_c$ such that $V_{c,j}^{\bar{\pi}}(s) \geq \tau$.

Then, strong duality holds and we can use the dual variable $\tilde{\lambda}_j^\alpha \in \text{argmin}_{\lambda_j\geq 0}\max_{\pi_\alpha\in\Pi_c} V_{r,j}^{\pi_\alpha}(s) + \lambda_j\left(V_{c,j}^{\pi_\alpha}(s) - \tau\right)$ to approximate the optimal Lagrange multiplier. The value of $\tilde{\lambda}_j^\alpha$ can be obtained by optimization with alternative updates:

$$i^{(t+1)} = \underset{i\in[N]}{\text{argmax}}\, V_{r,j}^{\tilde{\pi}_i}(s) + \lambda^{(t)}\left(V_{c,j}^{\tilde{\pi}_i}(s) - \tau\right) \tag{10}$$

$$\lambda^{(t+1)} = \mathrm{P}_{\mathbb{R}_{\geq 0}}\left(\lambda^{(t)} - \eta^{(t)}\left(V_{r,j}^{\tilde{\pi}_{i^{(t+1)}}}(s) - \tau\right)\right), \tag{11}$$

where $\mathrm{P}_{\mathbb{R}_{\geq 0}}$ projects an input onto the non-negative orthant. When the step size $\eta^{(t)}$ is chosen properly, this update process converges to the optimal solution,

Proposition 2. *If the sequence of step sizes $\left\{\eta^{(t)}\right\}_{t=1}^{\infty}$ satisfy $\eta^{(t)} \to 0$, $\sum_{t=1}^{\infty}\eta^{(t)} = \infty$ and $\sum_{t=1}^{\infty}\left[\eta^{(t)}\right]^2 \leq \infty$, then updates given in Eq. (10) and (11) will derive a dual variable that converges to the optimal dual variable, i.e., $\lambda^{(t)} \to \tilde{\lambda}_j^\alpha$.*

Algorithm and practical use. Our dual estimation method is summarized in Algorithm 1. The true value function of a source policy $\tilde{\pi}_i$ typically is not available in practice and has to be estimated, e.g., with a sample average by evaluating $\tilde{\pi}_i$'s SFs with multiple trajectories on the source task and using the target task's reward vectors. If the estimate $\hat{V}_{r,j}^{\tilde{\pi}_i}(s) = {}^1/_{K_i}\sum_{k=1}^{K_i} R_{r,j}(s,\omega_{i,k})$ is computed with a random sample of trajectories $\{\omega_{i,k}\}_{k=1}^{K_i}$ such that $\mathbb{E}\left[R_{r,j}(s,\omega_{i,k})\right] = V_{r,j}^{\tilde{\pi}_i}(s)\,\forall k$ for both r and c, then $\hat{\lambda}_j^\alpha$ can be shown to be a consistent estimator,

Proposition 3. *Denote by Λ_j the set of all optimal dual variable $\tilde{\lambda}_j^\alpha$. Let the estimator $\hat{\lambda}_j^\alpha \in \text{argmin}_{\lambda_j\geq 0}\max_{\pi_\alpha\in\Pi_c}\hat{V}_{r,j}^{\pi_\alpha}(s) + \lambda_j\left(\hat{V}_{c,j}^{\pi_\alpha}(s) - \tau\right)$. Then $\text{dist}\left(\hat{\lambda}_j^\alpha, \Lambda_j\right) \to 0$ with probability 1 as the sample size K_i, $i = 1, \cdots, N$, tends to infinity.*

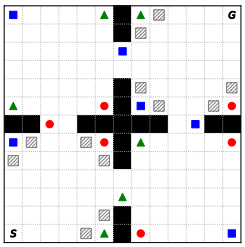
Once we have $\lambda^{(T)}$, we can select actions on the target task by $\pi(s) \in \text{argmax}_a \max_{i\in[N]} \tilde{Q}_{r,j}^{\tilde{\pi}_i}(s,a) + \lambda^{(T)}\tilde{Q}_{c,j}^{\tilde{\pi}_i}(s,a)$, where the transfer policy is given according to Eq. (5). Note that the optimal dual variable defined in Eq. (8) is for a specific state s. Therefore, dual estimation is repeatedly executed during roll-out on the target task. However, using Algorithm 1 on every state is computationally inefficient if the action-value functions for adjacent states are similar. In such cases, dual variable estimation can be performed periodically, which we found to work well in our experiments.

IV. Related work

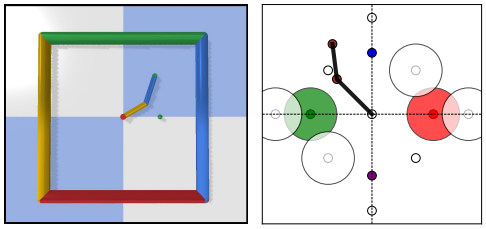
Our work is related to existing research in transfer and safety in reinforcement learning. We give a brief overview of recent work and refer readers desiring more comprehensive information to excellent survey articles [12], [13], [2]. Safety is a consideration not restricted to RL; various methods have been proposed in the machine learning, control, and robotics communities [14]. For example, popular control approaches include the design of Lyapunov functions [15] and barrier functions [16]. However, these techniques often assume prior knowledge of the environment dynamics and thus, are unsuitable for complex RL tasks where a known model of the environment is unavailable.

Transfer learning in RL. Knowledge transfer in RL can be achieved through various techniques, *e.g.*, via instances [17], parameters [18], skill compositions for new tasks [19], [20], [21], [22], and representations. Of particular relevance to our work are state representations that are informative of the reward [23]. The successor features (SFs) [5] employed in our work is an example of such state representation transfer. SFs generalize the successor representation [24] and decouple environmental dynamics from rewards. This enables policy transfer across tasks with theoretical performance guarantees (provided by GPI). However, its effectiveness relies on useful SFs that have to be learnt or provided under the same transition, limiting its applications in tasks with changing dynamics. For example, a navigation agent that avoids task-dependent obstacles or a robotic arm that manipulates different objects require quick learning of new dynamics. To address this limitation, Abdolshah *et al.* [25] model SFs with Gaussian Processes and show improved performance. Nevertheless, the assumption of noisy measurement can increase the upper bounded error. Recently, there has been significant work on extending SFs, *e.g.*, learn continuous control policy [26], [27], exploit task descriptions [28] and discover efficient paths [29]. These advances, along with improvements in the construction of reward features [30] and tasks with optimistic linear support [31], can potentially be used to enhance SFT-CoP's performance.

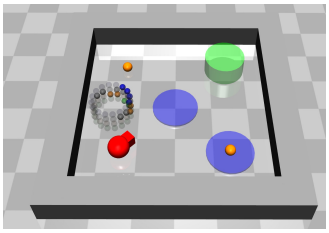
Safety in RL. The term "safety" is overloaded in the field of RL and here, we focus on learning in a hazardous



(a) Four-Room: 'S'/'G' is the start/goal location. ■, ▲ and ● are 3 types of objects with varying reward values on different tasks. ▨ are traps.



(b) Reacher. Left: a simulated two-link robotic arm based on MuJoCo physics engine. Right: four goal locations for training (colored filled) and eight for testing (white hollow) are represented by dots. Unsafe regions are plot in shaded circles.



(c) SafetyGym: The red agent has to reach the goal (green cylinder) while avoiding the blue unsafe regions. Orange points are buttons with varying reward values on different tasks.

Fig. 1: Benchmark domains used in our experiments.

environment in an online fashion with transfer learning. One popular approach is to minimize risk during agent's interaction with the environment where the occurrence of rare dangerous events can cause irreparable harm to the agent and/or others. Methods that are designed to capture this risk include policy optimization over the worst case return [32], variance of the return [33] and the conditional value at risk [34]. However, since these techniques optimize a scalar return, practical usage typically entails careful design to trade-off reward and risk.

An alternative approach—the one adopted in our work—is to model safety requirements as constraints. This results in a CMDP [3] for which various solvers have been developed. For example, with primal-dual optimization, [35] and [36] incorporate actor-critic for policy learning. Other methods, such as CPO [37], PCPO [38] and [39], aim to achieve near-constraint satisfaction at each training iteration. CMDPs are also related to multi-objective learning, but emphasize avoidance of constraint violations [40], [41]. In our work, we leverage the primal-dual approach for CMDPs, which enables SFs based policy transfer through the dual variable. For optimizing the policy, SFT-CoP employs standard Q-learning; recent advances [42], [43], [44] are complementary and may further improve upon our results.

Safety and transfer in RL. Methods that combine both transfer and safety in RL have only been explored in very recent work. For example, recent work [45] fine-tunes pre-trained policies on a CMDP using a combination of primal-dual optimization and Soft Actor-Critic [46]. However, this method lacks a means of efficient policy evaluation and therefore, has difficulties scaling to many tasks. The closest related work to ours is RaSFQL [4], which employs transfer via successor feature for risk-sensitive RL. RaSFQL learns robust policies that avoid high variance behavior, but suffers from the limitations (mentioned above) associated with methods that incorporate a risk-sensitive criterion to the objective. Despite limitations, the overall direction of these prior works is promising—risk is minimized via both a safety mechanism and sample complexity reduction. SFT-CoP is a step along this direction and the first to investigate efficient task transfer in the constrained setting.

V. Experiments

In this section we report on experiments designed to evaluate our primary hypothesis that SFT-CoP achieves positive transfer in terms of task accomplishment and, more importantly, constraints. We also sought to empirically evaluate the importance of the dual-variable during transfer.

Environments. We evaluated methods using three simulated environments (Fig. 1). The first two (Four-Room and Reacher) are benchmarks used in prior work [5], [4] to evaluate transfer learning in RL and safety. This last domain is a customized SafetyGym [9] environment where a robot (red agent in Fig. 1c) has to navigate to a goal location in a room. Please see the Appendix for implementation details.

Compared methods. We compared SFT-CoP against state-of-the-art transfer learning baselines: (1) Successor Feature Q-Learning (SFQL) [5], which performs transfer using GPI and SFs, but without considering constraints; the cost at each time step is added to the reward and hence, part of the return. (2) Risk-Aware SF Q-Learning (RaSFQL) [4], a state-of-the-art method safe policy transfer method that employs a risk-sensitive criterion. We also compared safe RL method CPO, and Primal-Dual Q-Learning (PDQL) which can be used to solve CMDPs with safety constraints. Comparing SFT-CoP against PDQL/CPO enable us to ascertain the value of transfer learning, while performance differences from SFQL and RaSFQL would be informative of any safety gains afforded by the constraint formulation. For the tasks where states and actions are continuous, we adopt the approach in prior work [4] by employing Deep Q-Networks (DQNs) [5], [1] and discretizing the action space.

A. Results and analysis

In the following section, we describe our key results, with extra results in the supplementary material. In general, we find that SFT-CoP provides stricter adherence to safety constraints compared to the baselines. Our experiments further emphasize the importance of the dual variable λ during transfer, which supports our theoretical findings.

Does SFT-CoP achieve positive transfer (in terms of task objectives and constraints) in safety-constrained settings? In brief, yes. Fig. 2 compares SFT-CoP against non-

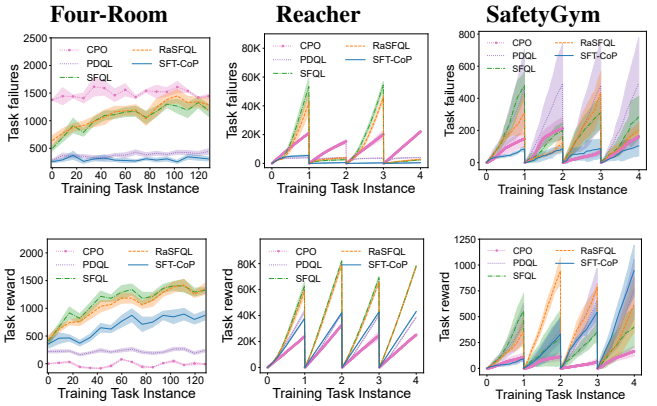


Fig. 2: Performance of CPO, PDQL, SFQL, RASFQL ($\beta = 2$) and SFT-COP on the Four-Room, Reacher and SafetyGym domains. We compare task failures (top row) and rewards (bottom row) between the different transfer methods over the training task instances.

transfer methods CPO and PDQL, and SFT-COP significantly outperform them. Without transferring from previous tasks, these methods do not converge and perform poorly, especially in later tasks. In contrast, transfer methods such as SFT-COP leverage accumulated knowledge to achieve larger rewards and fewer failures in later tasks.

How does SFT-COP compare against transfer RL methods? Fig. 2 compares the SFQL, RASFQL, and SFT-COP. During learning, exploration is conducted so failures are not completely avoidable, even with transfer. All methods attain increasing rewards as more tasks are seen, but SFT-COP generally achieves the lowest failures. Qualitatively, from Fig. 3 we observe that unlike SFT-COP, SFQL and RASFQL do not avoid the unsafe objects, e.g., in the bottom-left and top-right rooms in Foor-Rooms, or those within unsafe regions in Reacher/SafetyGym.

It should be noted that SFT-COP and RASFQL are motivated by different underlying principles—SFT-COP attempts to ensure compliance to constraints, while RASFQL seeks to reduce low-probability but high-cost failures. However, the standard reward variance formulation used in RASFQL tends to avoid not only uncertain cost, but also uncertain *reward*; Fig. 4 shows this phenomena empirically using the Four-Room domain modified with objects that provide positive reward with probability 5%. This problem does not occur with SFT-COP. Constraints can also be applied to avoid low probability unsafe events; when the unsafe regions have a low probability of failure (5% and 3.5%) [4], SFT-COP provides a similar level of safety as RASFQL.

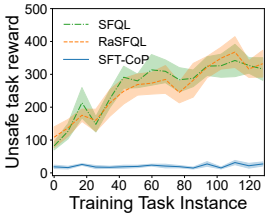


Fig. 3: Unsafe rewards averaged over training tasks of SFQL, RASFQL ($\beta = 2$) and SFT-COP on the Four-Room domain.

For SFT-COP, does accurate estimation of λ matter in practice? Our theoretical analysis suggests the importance

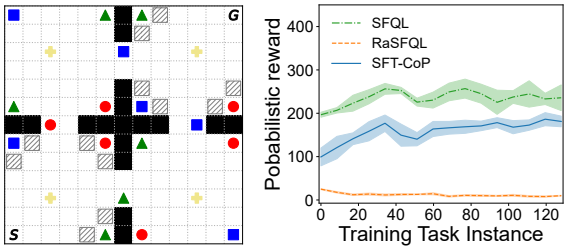


Fig. 4: Four-Room with probabilistic rewards from ✚ which RASFQL does not collect.

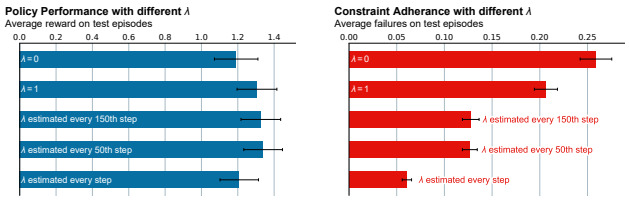


Fig. 5: Policy performance (left) and adherence to constraints (right) during transfer with different dual variables λ in the Four-Room domain.

of correctly estimating the optimal dual variable λ when transferring to a new task. The bar charts in Fig. 5 summarize the averaged rewards and failures of each episode with different methods of selecting $\tilde{\lambda}_j$. Setting $\tilde{\lambda}_j = 0$ corresponds to the case where the cost is ignored, and $\tilde{\lambda}_j = 1$ represents the setting where the cost is simply added to the reward. We also compare three different dual estimation update frequencies. As expected, the results show that estimating λ via Alg. 1 resulted in fewer failures compared to fixed $\lambda = \{0, 1\}$. Less frequent estimation led to more failures, but constraint adherence remains significantly better than the fixed λ variants.

VI. Conclusions and Future Work

In this paper, we presented SFT-COP, a principled method for constrained policy transfer in reinforcement learning using successor features. Unlike previous approaches, our approach separates out safety considerations as constraints. To our knowledge, this is the first work showing efficient transfer learning of constrained RL policies. Moving forward, there is a rich literature on optimization and solvers for CMDPs and a study into which methods work best alongside successor features could lead to performance improvements. Moreover, SFT-COP is limited to a discrete action space. It is possible to develop continuous policies under the maximum entropy RL framework [26], [27], while it remains an open question whether learning with primal dual can offer a similar guarantee. Our bound in Prop. 1 brings new insights to the dual variable, but shares similar properties as in [5] and [4] in terms of approximation error and scaling. Improving this bound would be interesting future work.

VII. Acknowledgements

This research is supported in part by the National Research Foundation (NRF), Singapore and DSO National Laboratories under the AI Singapore Program (Award Number: AISG2-RP-2020-017).

REFERENCES

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

[2] J. García and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Mach. Learning Research*, vol. 16, no. 42, p. 1437–1480, 2015.

[3] E. Altman, *Constrained Markov Decision Processes*. CRC Press, 1999, vol. 7.

[4] M. Gimelfarb, A. Barreto, S. Sanner, and C.-G. Lee, "Risk-aware transfer in reinforcement learning using successor features," in *NeurIPS*, 2021.

[5] A. Barreto, W. Dabney, R. Munos, J. J. Hunt, T. Schaul, H. P. van Hasselt, and D. Silver, "Successor features for transfer in reinforcement learning," in *NeurIPS*, vol. 30. Curran Associates, Inc., 2017.

[6] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st ed. USA: John Wiley & Sons, Inc., 1994.

[7] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[8] S. Paternain, M. Calvo-Fullana, L. F. O. Chamon, and A. Ribeiro, "Safe policies for reinforcement learning via primal-dual methods," *arXiv:1911.09101*, 2022.

[9] A. Ray, J. Achiam, and D. Amodei, "Benchmarking safe exploration in deep reinforcement learning," *arXiv:1910.01708*, 2019.

[10] A. Barreto, D. Borsa, J. Quan, T. Schaul, D. Silver, M. Hessel, D. Mankowitz, A. Zidek, and R. Munos, "Transfer in deep reinforcement learning using successor features and generalised policy improvement," in *Proc. of the 35th Intl. Conf. on Mach. Learning*, vol. 80. PMLR, 10–15 Jul 2018, pp. 501–510.

[11] S. Paternain, L. Chamon, M. Calvo-Fullana, and A. Ribeiro, "Constrained reinforcement learning has zero duality gap," in *NeurIPS*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[12] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *Journal of Mach. Learning Research*, vol. 10, no. 56, pp. 1633–1685, 2009.

[13] Z. Zhu, K. Lin, and J. Zhou, "Transfer learning in deep reinforcement learning: A survey," *arXiv:2009.07888*, 2020.

[14] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *arXiv:2108.06266*, 2021.

[15] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, "Safe model-based reinforcement learning with stability guarantees," in *NeurIPS*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[16] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," *Proc. of the AAAI Conf. on Artificial Intelligence*, vol. 33, no. 01, pp. 3387–3395, Jul. 2019.

[17] A. Lazaric, M. Restelli, and A. Bonarini, "Transfer of samples in batch reinforcement learning," in *Proc. of the 25th Intl. Conf. on Mach. Learning*, ser. ICML '08. New York, NY, USA: Association for Computing Mach.ry, 2008, p. 544–551.

[18] J. Rajendran, A. Srinivas, M. M. Khapra, P. Prasanna, and B. Ravindran, "Attend, adapt and transfer: Attentive deep architecture for adaptive transfer from multiple sources in the same domain," in *Intl. Conf. on Learning Representations*, 2017.

[19] P. Vaezipoor, A. C. Li, R. A. T. Icarte, and S. A. Mcilraith, "Ltl2action: Generalizing ltl instructions for multi-task rl," in *Proc. of the 38th Intl. Conf. on Mach. Learning*, ser. Proc. of Mach. Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 10 497–10 508.

[20] B. Araki, X. Li, K. Vodrahalli, J. Decastro, M. Fry, and D. Rus, "The logical options framework," in *Proc. of the 38th Intl. Conf. on Mach. Learning*, vol. 139. PMLR, 18–24 Jul 2021, pp. 307–317.

[21] E. Todorov, "Compositionality of optimal control laws," in *NeurIPS*, Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, Eds., vol. 22. Curran Associates, Inc., 2009.

[22] G. Nangue Tasse, S. James, and B. Rosman, "A boolean task algebra for reinforcement learning," in *NeurIPS*, vol. 33. Curran Associates, Inc., 2020, pp. 9497–9507.

[23] G. Konidaris and A. Barto, "Autonomous shaping: Knowledge transfer in reinforcement learning," in *Proc. of the 23rd Intl. Conf. on Mach. Learning*, 2006, p. 489–496.

[24] P. Dayan, "Improving generalization for temporal difference learning: The successor representation," *Neural Computation*, vol. 5, no. 4, pp. 613–624, 1993.

[25] M. Abdolshah, H. Le, T. K. George, S. Gupta, S. Rana, and S. Venkatesh, "A new representation of successor features for transfer across dissimilar environments," in *Proc. of the 38th Intl. Conf. on Mach. Learning*. PMLR, 18–24 Jul 2021, pp. 1–9.

[26] J. Hunt, A. Barreto, T. Lillicrap, and N. Heess, "Composing entropic policies using divergence correction," in *Proc. of the 36th Intl. Conf. on Mach. Learning*, ser. Proc. of Mach. Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 2911–2920.

[27] M. Mozifian, D. Fox, D. Meger, F. Ramos, and A. Garg, "Generalizing successor features to continuous domains for multi-task learning," 2022. [Online]. Available: https://openreview.net/forum?id=0m4c9ZfDrDt

[28] C. Ma, D. R. Ashley, J. Wen, and Y. Bengio, "Universal successor features for transfer reinforcement learning," *arXiv:2001.04025*, 2020.

[29] T. Moskovitz, S. R. Wilson, and M. Sahani, "A first-occupancy representation for reinforcement learning," in *Intl. Conf. on Learning Representations*, 2022.

[30] S. Alver and D. Precup, "Constructing a good behavior basis for transfer using generalized policy updates," *arXiv:2112.15025*, 2021.

[31] L. N. Alegre, A. Bazzan, and B. C. D. Silva, "Optimistic linear support and successor features as a basis for optimal policy transfer," in *Proc. of the 39th Intl. Conf. on Mach. Learning*, ser. Proc. of Mach. Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 394–413.

[32] M. Heger, "Consideration of risk in reinforcement learning," in *Mach. Learning Proceedings 1994*, W. W. Cohen and H. Hirsh, Eds. San Francisco (CA): Morgan Kaufmann, 1994, pp. 105–111.

[33] R. A. Howard and J. E. Matheson, "Risk-sensitive markov decision processes," *Management Science*, vol. 18, no. 7, pp. 356–369, 1972.

[34] A. Tamar, Y. Glassner, and S. Mannor, "Policy gradients beyond expectations: Conditional value-at-risk," *arXiv:1404.3862*, 2014.

[35] V. Borkar, "An actor-critic algorithm for constrained markov decision processes," *Systems & Control Letters*, vol. 54, no. 3, pp. 207–213, 2005.

[36] C. Tessler, D. J. Mankowitz, and S. Mannor, "Reward constrained policy optimization," in *Intl. Conf. on Learning Representations*, 2019.

[37] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proc. of the 34th Intl. Conf. on Mach. Learning*, vol. 70. PMLR, 06–11 Aug 2017, pp. 22–31.

[38] T.-Y. Yang, J. Rosca, K. Narasimhan, and P. J. Ramadge, "Projection-based constrained policy optimization," in *Intl. Conf. on Learning Representations*, 2020.

[39] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh, "A lyapunov-based approach to safe reinforcement learning," in *NeurIPS*, vol. 31. Curran Associates, Inc., 2018.

[40] S. Miryoosefi, K. Brantley, H. Daume III, M. Dudik, and R. E. Schapire, "Reinforcement learning with convex constraints," in *NeurIPS*, vol. 32, 2019.

[41] A. Gattami, Q. Bai, and V. Agarwal, "Reinforcement learning for multi-objective and constrained markov decision processes," *arXiv:1901.08978*, 2019.

[42] H. Le, C. Voloshin, and Y. Yue, "Batch policy learning under constraints," in *Proc. of the 36th Intl. Conf. on Mach. Learning*, ser. Proc. of Mach. Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 3703–3712.

[43] D. Ding, K. Zhang, T. Basar, and M. Jovanovic, "Natural policy gradient primal-dual method for constrained markov decision processes," in *NeurIPS*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 8378–8390.

[44] H. Wei, X. Liu, and L. Ying, "A provably-efficient model-free algorithm for constrained markov decision processes," *arXiv:2106.01577*, 2021.

[45] E. Knight and J. Achiam, "Safely transferring to unsafe environments with constrained reinforcement learning," in *Intl. Conf. on Learning Representations Workshop: Beyond 'tabula rasa' in reinforcement learning (BeTR-RL)*, 2020.

[46] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *arXiv:1801.01290*, 2018.