
State-wise Constrained Policy Optimization

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 Reinforcement Learning (RL) algorithms have shown tremendous success in simu-
2 lation environments, but their application to real-world problems faces significant
3 challenges, with safety being a major concern. In particular, enforcing state-wise
4 constraints is essential for many challenging tasks such as autonomous driving
5 and robot manipulation. However, existing safe RL algorithms under the frame-
6 work of Constrained Markov Decision Process (CMDP) do not consider state-wise
7 constraints. To address this gap, we propose State-wise Constrained Policy Opti-
8 mization (SCPO), the first general-purpose policy search algorithm for state-wise
9 constrained reinforcement learning. SCPO provides guarantees for state-wise con-
10 straint satisfaction in expectation. In particular, we introduce the framework of
11 Maximum Markov Decision Process, and prove that the worst-case safety violation
12 is bounded under SCPO. We demonstrate the effectiveness of our approach on
13 training neural network policies for extensive robot locomotion tasks, where the
14 agent must satisfy a variety of state-wise safety constraints. Our results show
15 that SCPO significantly outperforms existing methods and can handle state-wise
16 constraints in high-dimensional robotics tasks.

17 1 Introduction

18 Reinforcement learning (RL) has achieved remarkable progress in games and control tasks [Mnih
19 et al., 2015, Vinyals et al., 2019, Brown and Sandholm, 2018, He et al., 2022, Zhao et al., 2019].
20 However, one major barrier that limits the application of RL algorithms to real-world problems is
21 the lack of safety assurance. RL agents learn to make reward-maximizing decisions, which may
22 violate safety constraints. For example, an RL agent controlling a self-driving car may receive high
23 rewards by driving at high speeds but will be exposed to high chances of collision. Although the
24 reward signals can be designed to penalize risky behaviors, there is no guarantee for safety. In other
25 words, RL agents may sometimes prioritize maximizing the reward over ensuring safety, which can
26 lead to unsafe or even catastrophic outcomes [Gu et al., 2022].

27 Emerging in the literature, safe RL aims to provide safety guarantees during or after training. Early
28 attempts have been made under the framework of constrained Markov Decision Process, where the
29 majority of works enforce cumulative constraints or chance constraints [Ray et al., 2019, Achiam
30 et al., 2017a, Liu et al., 2021]. In real-world applications, however, many critical constraints are
31 instantaneous. For instance, collision avoidance must be enforced at all times for autonomous
32 cars [Zhao et al., 2023]. Another example is that when a robot holds a glass, the robot can only
33 release the glass when the glass is on a stable surface. The violation of those constraints will lead to
34 irreversible failures of the task. In this work, we focus on state-wise (instantaneous) constraints.

35 The State-wise Constrained Markov Decision Process (SCMDP) is a novel formulation in reinforce-
36 ment learning that requires policies to satisfy hard state-wise constraints. Unlike cumulative or
37 probabilistic constraints, state-wise constraints demand full compliance at each time step as for-
38 malized by [Zhao et al., 2023]. Existing state-wise safe RL methods can be categorized based on

39 whether safety is ensured during training. There is a fundamental limitation that it is impossible to
40 guarantee hard state-wise safety during training without prior knowledge of the dynamic model. The
41 best we can achieve in a model free setting is to learn to satisfy the constraints using as few samples
42 as possible, which is the focus of this paper. We aim to provide theoretical guarantees on state-wise
43 safety violation and worst case reward degradation during training.

44 Our approach is underpinned by a key insight that constraining the maximum violation is equivalent
45 to enforcing state-wise safety. This insight leads to a novel formulation of MDP called the *Maximum*
46 *Markov Decision Process* (MMDP). With MMDP, we establish a new theoretical result that provides
47 a bound on the difference between the maximum cost of two policies for episodic tasks. This result
48 expands upon the cumulative discounted reward and cost bounds for policy search using trust regions,
49 as previously documented in literature [Achiam et al., 2017b]. We leverage this result to design a
50 policy improvement step that not only guarantees worst-case performance degradation but also ensures
51 state-wise cost constraints. Our proposed algorithm, *State-wise Constrained Policy Optimization*
52 (SCPO), approximates the theoretically-justified update, which achieves a state-of-the-art trade-off
53 between safety and performance. Through experiments, we demonstrate that SCPO effectively
54 trains neural network policies with thousands of parameters on high-dimensional simulated robot
55 locomotion tasks; and is able to optimize rewards while enforcing state-wise safety constraints. This
56 work represents a significant step towards developing practical safe RL algorithms that can be applied
57 to many real-world problems.

58 2 Related Work

59 2.1 Cumulative Safety

60 Cumulative safety requires that the expected discounted return with respect to some cost function is
61 upper-bounded over the entire trajectory. One representative approach is constrained policy optimiza-
62 tion (CPO) [Achiam et al., 2017a], which builds on a theoretical bound on the difference between
63 the costs of different policies and derives a policy improvement procedure to ensure constraints
64 satisfaction. Another approach is interior-point policy optimization (IPO) [Liu et al., 2019], which
65 augments the reward-maximizing objective with logarithmic barrier functions as penalty functions
66 to accommodate the constraints. Other methods include Lagrangian methods [Ray et al., 2019]
67 which use adaptive penalty coefficients to enforce constraints and projection-based constrained
68 policy optimization (PCPO) [Yang et al., 2020a] which projects trust-region policy updates onto the
69 constraint set. Although our focus is on a different setting of constraints, existing methods are still
70 valuable references for illustrating the advantages of our SCPO. By utilizing MMDP, SCPO breaks
71 the conventional safety-reward trade-off, which results in stronger convergence of state-wise safety
72 constraints and guaranteed performance degradation bounds.

73 2.2 State-wise Safety

74 **Hierarchical Policy** One way to enforce state-wise safety constraints is to use hierarchical policies,
75 with an RL policy generating reward-maximizing actions, and a safety monitor modifying the actions
76 to satisfy state-wise safety constraints. Such an approach often requires a perfect safety critic to
77 function well. For example, conservative safety critics (CSC) [Bharadhwaj et al., 2020] propose
78 a safe critic $Q_C(s, a)$, providing a conservative estimate of the likelihood of being unsafe given a
79 state-action pair. If the safety violation exceeds a predefined threshold, a new action is re-sampled
80 from the policy until it passes the safety critic. However, this approach is time-consuming. On
81 the other hand, optimization-based methods such as gradient descent or quadratic programming
82 can be used to find a safe action that satisfies the constraint while staying close to the reference
83 action. Unrolling safety layer (USL) [Zhang et al., 2022a] follows a similar hierarchical structure as
84 CSC but performs gradient descent on the reference action iteratively until the constraint is satisfied
85 based on learned safety critic $Q_C(s, a)$. Finally, instead of using gradient descent, Lyapunov-based
86 policy gradient (LPG) [Chow et al., 2019] and SafeLayer [Dalal et al., 2018] directly solve quadratic
87 programming (QP) to project actions to the safe action set induced by the linearized versions of some
88 learned critic $Q_C(s, a)$. All these approaches suffer from safety violations due to imperfect critic
89 $Q_C(s, a)$, while those solving QPs further suffer from errors due to the linear approximation of the
90 critic. To avoid those issues, we propose SCPO as an end-to-end policy which does not explicitly
91 maintain a safety monitor.

92 **End-to-End Policy** End-to-end policies maximize task rewards while ensuring safety at the same
93 time. Related work regarding state-wise safety after convergence has been explored recently. Some
94 approaches [Liang et al., 2018, Tessler et al., 2018] solve a primal-dual optimization problem to
95 satisfy the safety constraint in expectation. However, the associated optimization is hard in practice
96 because the optimization problem changes at every learning step. [Bohez et al., 2019] approaches
97 the same setting by augmenting the reward with the sum of the constraint penalty weighted by the
98 Lagrangian multiplier. Although claimed state-wise safety performance, the aforementioned methods
99 do not provide theoretical guarantee and fail to achieve near-zero safety violation in practice. [He
100 et al., 2023] proposes AutoCost to automatically find an appropriate cost function using evolutionary
101 search over the space of cost functions as parameterized by a simple neural network. It is empirically
102 shown that the evolved cost functions achieve near-zero safety violation, however, no theoretical
103 guarantee is provided, and extensive computation is required. FAC [Ma et al., 2021] does provide
104 theoretically guaranteed state-wise safety via parameterized Lagrange functions. However, FAC
105 relies on strong assumptions and performs poorly in practice. To resolve the above issues, we
106 propose SCPO as an easy-to-implement and theoretically sound approach with no prior assumptions
107 on the underlying safety functions.

108 3 Problem Formulation

109 3.1 Preliminaries

110 In this paper, we are especially interested in guaranteeing safety for episodic tasks, which falls within
111 in the scope of finite-horizon Markov Decision Process (MDP). An MDP is specified by a tuple
112 $(\mathcal{S}, \mathcal{A}, \gamma, R, P, \mu)$, where \mathcal{S} is the state space, and \mathcal{A} is the control space, $R : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the
113 reward function, $0 \leq \gamma < 1$ is the discount factor, $\mu : \mathcal{S} \mapsto \mathbb{R}$ is the initial state distribution, and
114 $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$ is the transition probability function. $P(s'|s, a)$ is the probability of transitioning
115 to state s' given that the previous state was s and the agent took action a at state s . A stationary
116 policy $\pi : \mathcal{S} \mapsto \mathcal{P}(\mathcal{A})$ is a map from states to a probability distribution over actions, with $\pi(a|s)$
117 denoting the probability of selecting action a in state s . We denote the set of all stationary policies by
118 Π . Subsequently, we denote π_θ as the policy that is parameterized by the parameter θ .

119 The standard goal for MDP is to learn a policy π that maximizes a performance measure $\mathcal{J}_0(\pi)$ which
120 is computed via the discounted sum of reward:

$$\mathcal{J}_0(\pi) = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^H \gamma^t R(s_t, a_t, s_{t+1}) \right], \quad (1)$$

121 where $H \in \mathbb{N}$ is the horizon, $\tau = [s_0, a_0, s_1, \dots]$, and $\tau \sim \pi$ is shorthand for that the distribution
122 over trajectories depends on $\pi : s_0 \sim \mu, a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, a_t)$.

123 3.2 State-wise Constrained Markov Decision Process

124 A constrained Markov Decision Process (CMDP) is an MDP augmented with constraints that restrict
125 the set of allowable policies. Specifically, CMDP introduces a set of cost functions, C_1, C_2, \dots, C_m ,
126 where $C_i : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$ maps the state action transition tuple into a cost value. Analogous to (1),
127 we denote

$$\mathcal{J}_{C_i}(\pi) = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^H \gamma^t C_i(s_t, a_t, s_{t+1}) \right] \quad (2)$$

128 as the cost measure for policy π with respect to cost function C_i . Hence, the set of feasible stationary
129 policies for CMDP is then defined as follows, where $d_i \in \mathbb{R}$:

$$\Pi_C = \{\pi \in \Pi \mid \forall i, \mathcal{J}_{C_i}(\pi) \leq d_i\}. \quad (3)$$

130 In CMDP, the objective is to select a feasible stationary policy π_θ that maximizes the performance
131 measure:

$$\max_{\pi} \mathcal{J}_0(\pi), \text{ s.t. } \pi \in \Pi_C. \quad (4)$$

132 In this paper, we are interested in a special type of CMDP where the safety specification is to persis-
 133 tently satisfy a hard cost constraint **at every step** (as opposed to cumulative costs over trajectories),
 134 which we refer to as *State-wise Constrained Markov Decision Process* (SCMDP). Like CMDP,
 135 SCMDP uses the set of cost functions C_1, C_2, \dots, C_m to evaluate the instantaneous cost of state
 136 action transition tuples. Unlike CMDP, SCMDP requires the cost for every state action transition to
 137 satisfy a hard constraint. Hence, the set of feasible stationary policies for SCMDP is defined as

$$\bar{\Pi}_C = \{\pi \in \Pi \mid \forall i, \mathbb{E}_{(s_t, a_t, s_{t+1}) \sim \tau, \tau \sim \pi} [C_i(s_t, a_t, s_{t+1})] \leq w_i\} \quad (5)$$

138 where $w_i \in \mathbb{R}$. Then the objective for SCMDP is to find a feasible stationary policy from $\bar{\Pi}_C$ that
 139 maximizes the performance measure. Formally,

$$\max_{\pi} \mathcal{J}_0(\pi), \text{ s.t. } \pi \in \bar{\Pi}_C \quad (6)$$

140 3.3 Maximum Markov Decision Process

141 Note that for (6), every state-action transition pair corresponds to a constraint, which is intractable to
 142 solve using conventional reinforcement learning algorithms. Our intuition is that, instead of directly
 143 constraining the cost of each possible state-action transition, we can constrain the expected maximum
 144 state-wise cost along the trajectory, which is much easier to solve. Following that intuition, we define
 145 a novel *Maximum Markov-Decision Process* (MMDP), which further extends CMDP via (i) a set of
 146 up-to-now maximum state-wise costs $\mathbf{M} \doteq [M_1, M_2, \dots, M_m]$ where $M_i \in \mathcal{M} \subset \mathbb{R}$, and (ii) a set
 147 of *cost increment* functions, D_1, D_2, \dots, D_m , where $D_i : (\mathcal{S}, \mathcal{M}^m) \times \mathcal{A} \times \mathcal{S} \mapsto [0, \mathbb{R}^+]$ maps the
 148 augmented state action transition tuple into a non-negative cost increment. We define the augmented
 149 state $\hat{s} = (s, \mathbf{M}) \in (\mathcal{S}, \mathcal{M}^m) \doteq \hat{\mathcal{S}}$, where $\hat{\mathcal{S}}$ is the augmented state space. Formally,

$$D_i(\hat{s}_t, a_t, \hat{s}_{t+1}) = \max\{C_i(s_t, a_t, s_{t+1}) - M_{it}, 0\}. \quad (7)$$

150 By setting $D_i(\hat{s}_0, a_0, \hat{s}_1) = C_i(s_0, a_0, s_1)$, we have $M_{it} = \sum_{k=0}^{t-1} D_i(\hat{s}_k, a_k, \hat{s}_{k+1})$ for $t \geq 1$.
 151 Hence, we define *expected maximum state-wise cost* (or D_i -return) for π :

$$\mathcal{J}_{D_i}(\pi) = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^H D_i(\hat{s}_t, a_t, \hat{s}_{t+1}) \right]. \quad (8)$$

152 Importantly, (8) is the key component of MMDP and differs our work from existing safe RL ap-
 153 proaches that are based on CMDP cost measure (2). With (8), (6) can be rewritten as:

$$\max_{\pi} \mathcal{J}(\pi), \text{ s.t. } \forall i, \mathcal{J}_{D_i}(\pi) \leq w_i, \quad (9)$$

154 where $\mathcal{J}(\pi) = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^H \gamma^t R(\hat{s}_t, a_t, \hat{s}_{t+1}) \right]$ and $R(\hat{s}, a, \hat{s}') \doteq R(s, a, s')$. With $R(\tau)$ being the
 155 discounted return of a trajectory, we define the on-policy value function as $V^\pi(\hat{s}) \doteq \mathbb{E}_{\tau \sim \pi} [R(\tau) \mid \hat{s}_0 =$
 156 $\hat{s}]$, the on-policy action-value function as $Q^\pi(\hat{s}, a) \doteq \mathbb{E}_{\tau \sim \pi} [R(\tau) \mid \hat{s}_0 = \hat{s}, a_0 = a]$, and the advantage
 157 function as $A^\pi(\hat{s}, a) \doteq Q^\pi(\hat{s}, a) - V^\pi(\hat{s})$. Lastly, we define on-policy value functions, action-value
 158 functions, and advantage functions for the cost increments in analogy to V^π , Q^π , and A^π , with D_i
 159 replacing R , respectively. We denote those by $V_{D_i}^\pi$, $Q_{D_i}^\pi$ and $A_{D_i}^\pi$.

160 4 State-wise Constrained Policy Optimization

161 To solve large and continuous MDPs, policy search algorithms search for the optimal policy within a
 162 set $\Pi_\theta \subset \Pi$ of parametrized policies. In local policy search [Peters and Schaal, 2008], the policy is
 163 iteratively updated by maximizing $\mathcal{J}(\pi)$ over a local neighborhood of the most recent policy π_k . In
 164 local policy search for SCMDPs, policy iterates must be feasible, so optimization is over $\Pi_\theta \cap \bar{\Pi}_C$.
 165 The optimization problem is:

$$\begin{aligned} \pi_{k+1} &= \underset{\pi \in \Pi_\theta}{\operatorname{argmax}} \mathcal{J}(\pi), \\ &\text{s.t. } \operatorname{Dist}(\pi, \pi_k) \leq \delta, \\ &\quad \mathcal{J}_{D_i}(\pi) \leq w_i, i = 1, \dots, m. \end{aligned} \quad (10)$$

166 where $Dist$ is some distance measure, and $\delta > 0$ is a step size. For actual implementation, we need
 167 to evaluate the constraints first in order to determine the feasible set. However, it is challenging to
 168 evaluate the constraints using samples during the learning process. In this work, we propose SCPO
 169 inspired by recent trust region optimization methods [Schulman et al., 2015]. SCPO approximates
 170 (10) using (i) KL divergence distance metric $Dist$ and (ii) surrogate functions for the objective and
 171 constraints, which can be easily estimated from samples on π_k . Mathematically, SCPO requires
 172 the policy update at each iteration is bounded within a trust region, and updates policy via solving
 173 following optimization:

$$\begin{aligned} \pi_{k+1} = \mathop{\text{argmax}}_{\pi \in \Pi_\theta} & \mathbb{E}_{\substack{\hat{s} \sim d^{\pi_k} \\ a \sim \pi}} [A^{\pi_k}(\hat{s}, a)] & (11) \\ \text{s.t.} & \mathbb{E}_{\hat{s} \sim \bar{d}^{\pi_k}} [\mathcal{D}_{KL}(\pi \| \pi_k)[\hat{s}]] \leq \delta, \\ & \mathcal{J}_{D_i}(\pi_k) + \mathbb{E}_{\substack{\hat{s} \sim \bar{d}^{\pi_k} \\ a \sim \pi}} \left[A_{D_i}^{\pi_k}(\hat{s}, a) \right] + 2(H+1)\epsilon_{D_i}^\pi \sqrt{\frac{1}{2}\delta} \leq w_i, i = 1, \dots, m. \end{aligned}$$

174 where $\mathcal{D}_{KL}(\pi' \| \pi)[\hat{s}]$ is KL divergence between two policy (π', π) at state \hat{s} , the set $\{\pi \in$
 175 $\Pi_\theta : \mathbb{E}_{\hat{s} \sim \bar{d}^{\pi_k}} [\mathcal{D}_{KL}(\pi \| \pi_k)[\hat{s}]] \leq \delta\}$ is called *trust region*, $d^{\pi_k} \doteq (1 - \gamma) \sum_{t=0}^H \gamma^t P(\hat{s}_t = \hat{s} | \pi_k)$,
 176 $\bar{d}^{\pi_k} \doteq \sum_{t=0}^H P(\hat{s}_t = \hat{s} | \pi_k)$ and $\epsilon_{D_i}^\pi \doteq \mathbf{max}_{\hat{s}} \mathbb{E}_{a \sim \pi} [A_{D_i}^{\pi_k}(\hat{s}, a)]$. We then show that SCPO guaran-
 177 tees (i) worst case maximum state-wise cost violation, and (ii) worst case performance degradation
 178 for policy update, by establishing new bounds on the difference in returns between two stochastic
 179 policies π and π' for MMDPs.

180 **Theoretical Guarantees for SCPO** We start with the theoretical foundation for our approach,
 181 i.e. a new bound on the difference in state-wise maximum cost between two arbitrary policies. The
 182 following theorem connects the difference in maximum state-wise cost between two arbitrary policies
 183 to the total variation divergence between them. Here total variation divergence between discrete
 184 probability distributions p, q is defined as $\mathcal{D}_{TV}(p \| q) = \frac{1}{2} \sum_i |p_i - q_i|$. This measure can be easily
 185 extended to continuous states and actions by replacing the sums with integrals. Thus, the total variation
 186 divergence between two policy (π', π) at state \hat{s} is defined as: $\mathcal{D}_{TV}(\pi' \| \pi)[\hat{s}] = \mathcal{D}_{TV}(\pi'(\cdot | \hat{s}) \| \pi(\cdot | \hat{s}))$.
 187 **Theorem 1** (Trust Region Update State-wise Maximum Cost Bound). *For any policies π', π , with*
 188 $\epsilon_{D_i}^{\pi'} \doteq \mathbf{max}_{\hat{s}} \mathbb{E}_{a \sim \pi'} [A_{D_i}^{\pi'}(\hat{s}, a)]$, and define $\bar{d}^\pi = \sum_{t=0}^H P(\hat{s}_t = \hat{s} | \pi)$ as the non-discounted aug-
 189 mented state distribution using π , then the following bound holds:

$$\mathcal{J}_D(\pi') - \mathcal{J}_D(\pi) \leq \mathbb{E}_{\substack{\hat{s} \sim \bar{d}^\pi \\ a \sim \pi'}} \left[A_{D_i}^{\pi'}(\hat{s}, a) + 2(H+1)\epsilon_{D_i}^{\pi'} \mathcal{D}_{TV}(\pi' \| \pi)[\hat{s}] \right]. \quad (12)$$

190 The proof for Theorem 1 is summarized in Appendix A. Next, we note the following relationship
 191 between the total variation divergence and the KL divergence [Boyd et al., 2003; Achiam et al., 2017a]:
 192 $\mathbb{E}_{\hat{s} \sim \bar{d}^\pi} [\mathcal{D}_{TV}(p \| q)[\hat{s}]] \leq \sqrt{\frac{1}{2} \mathbb{E}_{\hat{s} \sim \bar{d}^\pi} [\mathcal{D}_{KL}(p \| q)[\hat{s}]]}$. The following bound then follows directly from
 193 Theorem 1:

$$\mathcal{J}_D(\pi') \leq \mathcal{J}_D(\pi) + \mathbb{E}_{\substack{\hat{s} \sim \bar{d}^\pi \\ a \sim \pi'}} \left[A_{D_i}^{\pi'}(\hat{s}, a) + 2(H+1)\epsilon_{D_i}^{\pi'} \sqrt{\frac{1}{2} \mathbb{E}_{\hat{s} \sim \bar{d}^\pi} [\mathcal{D}_{KL}(\pi' \| \pi)[\hat{s}]]} \right]. \quad (13)$$

194 By Equation (13), we have a guarantee for satisfaction of maximum state-wise constraints:

195 **Proposition 1** (SCPO Update Constraint Satisfaction). *Suppose π_k, π_{k+1} are related by (11), then*
 196 *D_i -return for π_{k+1} satisfies*

$$\forall i, \mathcal{J}_{D_i}(\pi_{k+1}) \leq w_i.$$

197

198 Proposition 1 presents the first constraint satisfaction guarantee under MMDP. Unlike trust region
 199 methods such as CPO and TRPO, which assume a discounted sum characteristic, MMDP's non-
 200 discounted sum characteristic invalidates these theories. As the maximum state-wise cost is calculated

201 through a summation of non-discounted increments, analysis must be performed on a finite horizon to
 202 upper bound the worst-case summation. In contrast, the theory behind CPO relies on infinite horizon
 203 analysis with discounted constraint assumptions, which is not applicable for MMDP settings.

204 Next, we provide the performance guarantee of SCPO. Previous analyses of performance guarantees
 205 have focused on infinite-horizon MDP. We generalize the analysis to finite-horizon MDP, inspired
 206 by previous work [Kakade and Langford, 2002, Schulman et al., 2015, Achiam et al., 2017a], and
 207 prove it in Appendix B. The infinite-horizon case can be viewed as a special case of the finite-horizon
 208 setting.

209 **Proposition 2** (SCPO Update Worst Performance Degradation). *Suppose π_k, π_{k+1} are related by*
 210 *(11), with $\epsilon^{\pi_{k+1}} \doteq \max_{\hat{s}} \mathbb{E}_{a \sim \pi_{k+1}} [A^{\pi_k}(\hat{s}, a)]$, then performance return for π_{k+1} satisfies*

$$\mathcal{J}(\pi_{k+1}) - \mathcal{J}(\pi_k) \geq -\frac{\sqrt{2\delta}\gamma\epsilon^{\pi_{k+1}}}{1-\gamma}.$$

211 5 Practical Implementation

212 In this section, we show how to (a) implement an efficient approximation to the update (11), (b)
 213 encourage learning even when (11) becomes infeasible, and (c) handle the difficulty of fitting
 214 augmented value $V_{D_i}^{\pi}$ which is unique to our novel MMDP formulation. The full SCPO pseudocode
 215 is given as algorithm 1 in appendix C.

216 **Practical implementation with sample-based estimation** We first estimate the objective and
 217 constraints in (11) using samples. Note that we can replace the expected advantage on rewards using
 218 an importance sampling estimator with a sampling distribution π_k [Achiam et al., 2017a] as

$$\mathbb{E}_{\hat{s} \sim d^{\pi_k}, a \sim \pi} [A^{\pi_k}(\hat{s}, a)] = \mathbb{E}_{\hat{s} \sim d^{\pi_k}, a \sim \pi_k} \left[\frac{\pi(a|\hat{s})}{\pi_k(a|\hat{s})} A^{\pi_k}(\hat{s}, a) \right]. \quad (14)$$

219 (14) allows us to replace A^{π_k} with empirical estimates at each state-action pair (\hat{s}, a) from rollouts
 220 by the previous policy π_k . The empirical estimate of reward advantage is given by $R(\hat{s}, a, \hat{s}') +$
 221 $\gamma V^{\pi_k}(\hat{s}') - V^{\pi_k}(\hat{s})$. $V^{\pi_k}(\hat{s})$ can be computed at each augmented state by taking the discounted
 222 future return. The same can be applied to the expected advantage with respect to cost increments, with
 223 the sample estimates given by $D_i(\hat{s}, a, \hat{s}') + V_{D_i}^{\pi_k}(\hat{s}') - V_{D_i}^{\pi_k}(\hat{s})$. $V_{D_i}^{\pi_k}(\hat{s})$ is computed by taking the
 224 non-discounted future D_i -return. To proceed, we convexify (11) by approximating the objective and
 225 cost constraint via first-order expansions, and the trust region constraint via second-order expansions.
 226 Then, (11) can be efficiently solved using duality [Achiam et al., 2017a].

227 **Infeasible constraints** An update to θ is computed every time (11) is solved. However, due to
 228 approximation errors, sometimes (11) can become infeasible. In that case, we follow [Achiam
 229 et al., 2017a] to propose a recovery update that only decreases the constraint value within the trust
 230 region. In addition, approximation errors can also cause the proposed policy update (either feasible
 231 or recovery) to violate the original constraints in (11). Hence, each policy update is followed by
 232 a backtracking line search to ensure constraint satisfaction. If all these fails, we relax the search
 233 condition by also accepting decreasing expected advantage with respect to the costs, when the cost
 234 constraints are already violated. Denoting $c_i \doteq \mathcal{J}_{D_i}(\pi_k) + 2(H+1)\epsilon_D^{\pi} \sqrt{\delta/2} - w_i$, the above criteria
 235 can be summarized as

$$\mathbb{E}_{\hat{s} \sim \bar{d}^{\pi_k}} [\mathcal{D}_{KL}(\pi || \pi_k)[\hat{s}]] \leq \delta \quad (15)$$

$$\mathbb{E}_{\hat{s} \sim \bar{d}^{\pi_k}, a \sim \pi} [A_{D_i}^{\pi_k}(\hat{s}, a)] - \mathbb{E}_{\hat{s} \sim \bar{d}^{\pi_k}, a \sim \pi_k} [A_{D_i}^{\pi_k}(\hat{s}, a)] \leq \max(-c_i, 0). \quad (16)$$

236 Note that the previous expected advantage $\mathbb{E}_{\hat{s} \sim \bar{d}^{\pi_k}, a \sim \pi_k} [A_{D_i}^{\pi_k}(\hat{s}, a)]$ is also estimated from rollouts
 237 by π_k and converges to zero asymptotically, which recovers the original cost constraints in (11).

238 **Imbalanced cost value targets** A critical step in solving (11) is to fit the cost increment value
 239 functions $V_{D_i}^{\pi_k}(\hat{s}_t)$. By definition, $V_{D_i}^{\pi_k}(\hat{s}_t)$ is equal to the maximum cost increment in any future
 240 state over the maximal state-wise cost so far. In other words, the true $V_{D_i}^{\pi_k}$ will always be zero for all
 241 $\hat{s}_{t:H}$ when the maximal state-wise cost has already occurred before time t . In practice, this causes
 242 the distribution of cost increment value function to be highly zero-skewed and makes the fitting very
 243 hard. To mitigate the problem, we sub-sample the zero-valued targets to match the population of
 244 non-zero values. We provide more analysis on this trick in Q3 in section 6.2.

245 6 Experiments

246 In our experiments, we aim to answer these questions:

247 **Q1** How does SCPO compare with other state-of-the-
248 art methods for safe RL?

249 **Q2** What benefits are demonstrated by constraining
250 the maximum state-wise cost?

251 **Q3** How do the sub-sampling trick of SCPO impact
252 its performance?

253 6.1 Experiment Setups

254 **New Safety Gym** To showcase the effectiveness
255 of our state-wise constrained policy optimization
256 approach, we enhance the widely recognized safe rein-
257 forcement learning benchmark environment, Safety
258 Gym [Ray et al. [2019]], by incorporating additional
259 robots and constraints. Subsequently, we perform a
260 series of experiments on this augmented environment.

261 Our experiments are based on five different robots: (i)

262 **Point:** Figure 2a A point-mass robot ($\mathcal{A} \subseteq \mathbb{R}^2$) that

263 can move on the ground. (ii) **Swimmer:** Figure 2b

264 A three-link robot ($\mathcal{A} \subseteq \mathbb{R}^2$) that can move on the

265 ground. (iii) **Walker:** Figure 2d A bipedal robot

266 ($\mathcal{A} \subseteq \mathbb{R}^{10}$) that can move on the ground. (iv) **Ant:** Figure 2c A quadrupedal robot ($\mathcal{A} \subseteq \mathbb{R}^8$) that

267 can move on the ground. (v) **Drone:** Figure 2e A quadrotor robot ($\mathcal{A} \subseteq \mathbb{R}^4$) that can move in the air.

268 All of the experiments are based on the goal task where the robot must navigate to a goal. Additionally,
269 since we are interested in episodic tasks (finite-horizon MDP), the environment will be reset once the
270 goal is reached. For the robots that can move in 3D spaces (e.g, the Drone robot), we also design a
271 new 3D goal task with a sphere goal floating in the 3D space. Three different types of constraints are
272 considered: (i) **Hazard:** Dangerous areas as shown in Figure 3a. Hazards are trespassable circles on
273 the ground. The agent is penalized for entering them. (ii) **3D Hazard:** 3D Dangerous areas as shown
274 in Figure 3b. 3D Hazards are trespassable spheres in the air. The agent is penalized for entering them.
(iii) **Pillar:** Fixed obstacles as shown in Figure 3c. The agent is penalized for hitting them.

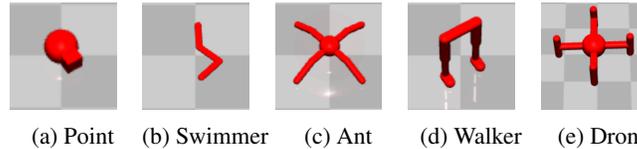


Figure 2: Robots for benchmark problems in upgraded Safety Gym.

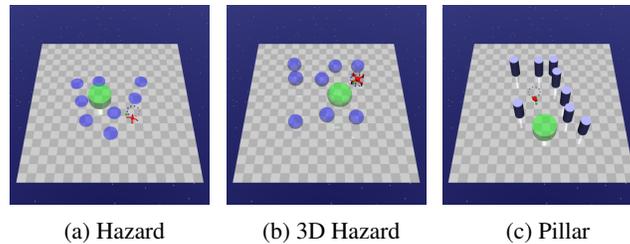


Figure 3: Constraints for benchmark problems in upgraded Safety Gym.

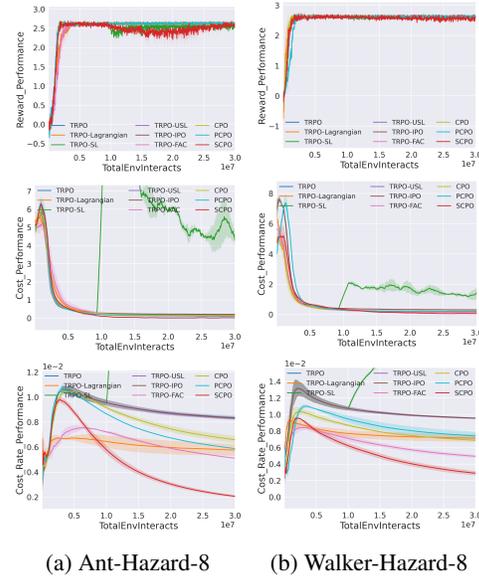


Figure 1: Comparison of results from two representative test suites in high dimensional systems (Ant and Walker).

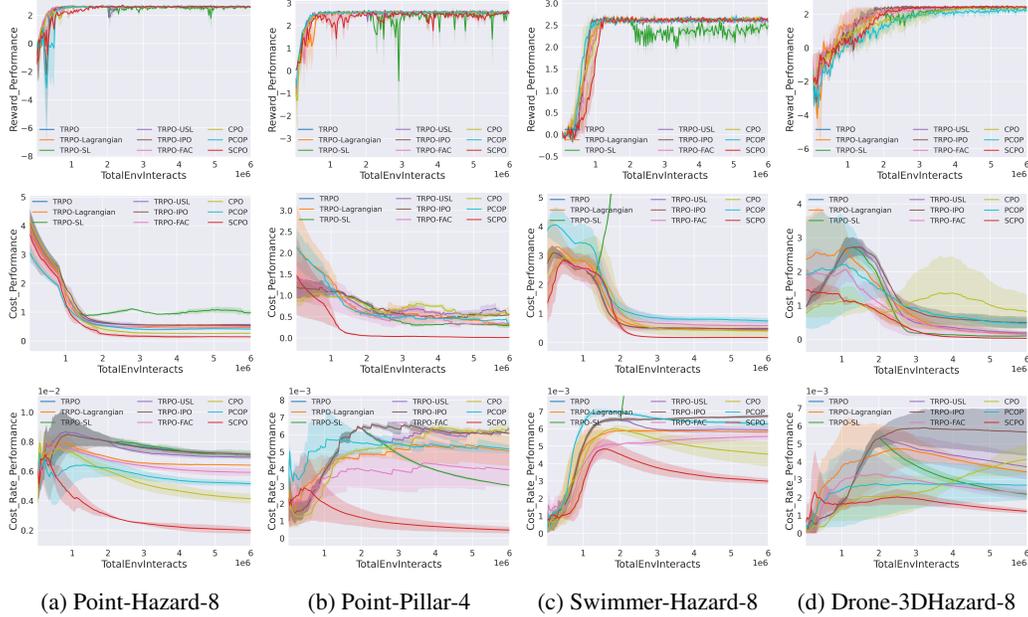


Figure 4: Comparison of results from four representative test suites in low dimensional systems (Point, Swimmer, and Drone).

276 Considering different robots, constraint types, and constraint difficulty levels, we design 14 test suites
 277 with 5 types of robots and 9 types of constraints, which are summarized in Table 1 in Appendix. We
 278 name these test suites as {Robot}-{Constraint Type}-{Constraint Number}.

279 **Comparison Group** The methods in the comparison group include: (i) unconstrained RL algorithm
 280 TRPO [Schulman et al., 2015] (ii) end-to-end constrained safe RL algorithms CPO [Achiam et al.,
 281 2017a], TRPO-Lagrangian [Bohez et al., 2019], TRPO-FAC [Ma et al., 2021], TRPO-IPO [Liu et al.,
 282 2020], PCPO [Yang et al., 2020b], and (iii) hierarchical safe RL algorithms TRPO-SL (TRPO-Safety
 283 Layer) [Dalal et al., 2018], TRPO-USL (TRPO-Unrolling Safety Layer) [Zhang et al., 2022b]. We
 284 select TRPO as our baseline method since it is state-of-the-art and already has safety-constrained
 285 derivatives that can be tested off-the-shelf. For hierarchical safe RL algorithms, we employ a warm-up
 286 phase (1/3 of the whole epochs) which does unconstrained TRPO training, and the generated data
 287 will be used to pre-train the safety critic for future epochs. For all experiments, the policy π , the value
 288 (V^π, V_D^π) are all encoded in feedforward neural networks using two hidden layers of size (64,64)
 289 with tanh activations. More details are provided in Appendix D.

290 **Evaluation Metrics** For comparison, we evaluate algorithm performance based on (i) reward
 291 performance, (ii) average episode cost and (iii) cost rate. Comparison metric details are provided
 292 in Appendix D.3. We set the limit of cost to 0 for all the safe RL algorithms since we aim to avoid
 293 any violation of the constraints. For our comparison, we implement the baseline safe RL algorithms
 294 exactly following the policy update / action correction procedure from the original papers. We
 295 emphasize that in order for the comparison to be fair, we give baseline safe RL algorithms every
 296 advantage that is given to SCPO, including equivalent trust region policy updates.

297 6.2 Evaluating SCPO and Comparison Analysis

298 **Low Dimension System** We select four representative test suites on low dimensional system
 299 (Point, Swimmer, Drone) and summarize the comparison results on Figure 4, which demonstrate
 300 that SCPO is successful at approximately enforcing zero constraints violation safety performance
 301 in all environments after the policy converges. Specifically, compared with the baseline safe RL
 302 methods, SCPO is able to achieve (i) near zero average episode cost and (ii) significantly lower
 303 cost rate without sacrificing reward performance. The baseline end-to-end safe RL methods (TRPO-
 304 Lagrangian, TRPO-FAC, TRPO-IPO, CPO, PCPO) fail to achieve the near zero cost performance

305 even when the cost limit is set to be 0. The baseline hierarchical safe RL methods (TRPO-SL,
 306 TRPO-USL) also fail to achieve near zero cost performance even with an explicit safety layer to
 307 correct the unsafe action at every time step. End-to-end safe RL algorithms fail since all methods
 308 rely on CMDP to minimize the discounted cumulative cost while SCPO directly work with MMDP
 309 to restrict the state-wise maximum cost by Proposition 1. We also observe that TRPO-SL fails to
 310 lower the violation during training, due to the fact that the linear approximation of cost function
 311 $C(\hat{s}_t, a, \hat{s}_{t+1})$ [Dalal et al., 2018] becomes inaccurate when the dynamics are highly nonlinear like
 312 the ones we used in MuJoCo [Todorov et al., 2012]. More detailed metrics for comparison and
 313 experimental results on test suites with low dimension systems are summarized in Appendix D.3.

314 **High Dimension System** To demonstrate the scalability and per-
 315 formance of SCPO in high-dimensional systems, we conducted additional tests on the Ant-Hazard-8 and Walker-Hazard-8
 316 suites, with 8-dimensional and 10-dimensional control spaces, respectively. The
 317 comparison results for high-dimensional systems are summarized in
 318 Figure 1, which show that SCPO outperforms all other baselines in
 319 enforcing zero safety violation without compromising performance
 320 in terms of return. SCPO rapidly stabilizes the cost return around
 321 zero and significantly reduces the cost rate, while the other baselines
 322 fail to converge to a policy with near-zero cost. The comparison
 323 results of both low dimension and high dimension systems answer
 324 **Q1**.
 325

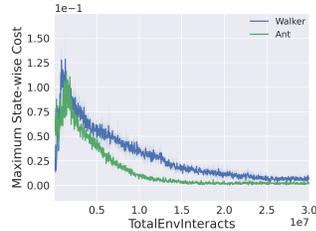


Figure 5:
Maximum state-wise cost

326 **Maximum State-wise Cost** As pointed in Section 3.3, the under-
 327 lying magic for enabling near-zero safety violation is to restrict the maximum state-wise cost to stay
 328 around zero. To have a better understanding of this process, we visualize the evolution of maximum
 329 state-wise cost for SCPO on the challenging high-dimensional Ant-Hazard-8 and Walker-Hazard-8
 330 test suites in Figure 5, which answers **Q2**.

331 **Ablation on Sub-sampling Imbalanced Cost Increment Value Targets** As pointed in Section 5, fitting $V_{D_i}^{\pi^k}(\hat{s}_t)$ is a critical step towards solving SCPO, which is challenging due to zero-skewed distribution of cost increment value function. To demonstrate the necessity of sub-sampling for solving this challenge, we compare the performance of SCPO with and without sub-sampling trick on the aerial robot test suite, summarized in Figure 6. It is evident that with sub-sampling, the agent achieves higher rewards and more importantly, converges to near-zero costs. That is because sub-sampling effectively balances the cost increment value targets and improves the fitting of $V_{D_i}^{\pi^k}(\hat{s}_t)$. We also attempted to solve the imbalance issue via over-sampling non-zero targets, but did not observe promising results. This ablation study provides insights into **Q3**.

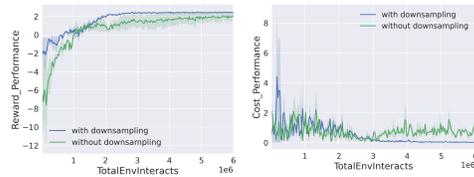


Figure 6: SCPO sub-sampling ablation study with Drone-3DHazard-8

345 7 Conclusion and Future Work

346 This paper proposed SCPO, the first general-purpose policy search algorithm for state-wise con-
 347 strained RL. Our approach provides guarantees for state-wise constraint satisfaction at each iteration,
 348 allows training of high-dimensional neural network policies while ensuring policy behavior, and is
 349 based on a new theoretical result on Maximum Markov Decision Process. We demonstrate SCPO's
 350 effectiveness on robot locomotion tasks, showing its significant performance improvement compared
 351 to existing methods and ability to handle state-wise constraints.

352 **Limitation and future work** One limitation of our work is that, although SCPO satisfies state-wise
 353 constraints, the theoretical results are valid only in expectation, meaning that constraint violations
 354 are still possible during deployment. To address that, we will study absolute state-wise constraint
 355 satisfaction, i.e. bounding the *maximal possible* state-wise cost, which is even stronger than the
 356 current result (satisfaction in expectation).

357 **References**

- 358 Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare,
359 Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control
360 through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.
- 361 Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung
362 Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in
363 starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.
- 364 Noam Brown and Tuomas Sandholm. Superhuman ai for heads-up no-limit poker: Libratus beats top
365 professionals. *Science*, 359(6374):418–424, 2018.
- 366 Tairan He, Yuge Zhang, Kan Ren, Minghuan Liu, Che Wang, Weinan Zhang, Yuqing Yang, and
367 Dongsheng Li. Reinforcement learning with automated auxiliary loss search. *arXiv preprint*
368 *arXiv:2210.06041*, 2022.
- 369 Wei-Ye Zhao, Xi-Ya Guan, Yang Liu, Xiaoming Zhao, and Jian Peng. Stochastic variance reduction
370 for deep q-learning. *arXiv preprint arXiv:1905.08152*, 2019.
- 371 Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, Yaodong Yang, and
372 Alois Knoll. A review of safe reinforcement learning: Methods, theory and applications. *arXiv*
373 *preprint arXiv:2205.10330*, 2022.
- 374 Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement
375 learning. *CoRR*, abs/1910.01708, 2019.
- 376 Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In
377 *International conference on machine learning*, pages 22–31. PMLR, 2017a.
- 378 Yongshuai Liu, Avishai Halev, and Xin Liu. Policy learning with constraints in model-free reinforce-
379 ment learning: A survey. In *The 30th International Joint Conference on Artificial Intelligence*
380 *(IJCAI)*, 2021.
- 381 Weiye Zhao, Tairan He, Rui Chen, Tianhao Wei, and Changliu Liu. State-wise safe reinforcement
382 learning: A survey. *The 32nd International Joint Conference on Artificial Intelligence (IJCAI)*,
383 2023.
- 384 Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In
385 *International Conference on Machine Learning*, pages 22–31. PMLR, 2017b.
- 386 Yongshuai Liu, Jiaxin Ding, and Xin Liu. IPO: interior-point policy optimization under constraints.
387 *CoRR*, abs/1910.09615, 2019. URL <http://arxiv.org/abs/1910.09615>.
- 388 Tsung-Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J. Ramadge. Projection-based
389 constrained policy optimization. *CoRR*, abs/2010.03152, 2020a. URL <https://arxiv.org/abs/2010.03152>.
- 390
- 391 Homanga Bharadhwaj, Aviral Kumar, Nicholas Rhinehart, Sergey Levine, Florian Shkurti, and
392 Animesh Garg. Conservative safety critics for exploration. *arXiv preprint arXiv:2010.14497*, 2020.
- 393 Linrui Zhang, Qin Zhang, Li Shen, Bo Yuan, Xueqian Wang, and Dacheng Tao. Evaluating model-free
394 reinforcement learning toward safety-critical tasks. *arXiv preprint arXiv:2212.05727*, 2022a.
- 395 Yinlam Chow, Ofir Nachum, Aleksandra Faust, Edgar Duenez-Guzman, and Mohammad
396 Ghavamzadeh. Lyapunov-based safe policy optimization for continuous control. *ICML 2019*
397 *Workshop RL4RealLife*, abs/1901.10031, 2019.
- 398 Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval
399 Tassa. Safe exploration in continuous action spaces. *CoRR*, abs/1801.08757, 2018.
- 400 Qingkai Liang, Fanyu Que, and Eytan Modiano. Accelerated primal-dual policy optimization for
401 safe reinforcement learning. *arXiv preprint arXiv:1802.06480*, 2018.
- 402 Chen Tessler, Daniel J Mankowitz, and Shie Mannor. *arXiv preprint arXiv:1805.11074*, 2018.

- 403 Steven Bohez, Abbas Abdolmaleki, Michael Neunert, Jonas Buchli, Nicolas Heess, and Raia Hadsell.
404 Value constrained model-free continuous control. *arXiv preprint arXiv:1902.04623*, 2019.
- 405 Tairan He, Weiye Zhao, and Changliu Liu. Autocost: Evolving intrinsic cost for zero-violation
406 reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.
- 407 Haitong Ma, Yang Guan, Shegnbo Eben Li, Xiangteng Zhang, Sifa Zheng, and Jianyu Chen. Feasible
408 actor-critic: Constrained reinforcement learning for ensuring statewise safety. *arXiv preprint*
409 *arXiv:2105.10682*, 2021.
- 410 Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients. *Neural*
411 *networks*, 21(4):682–697, 2008.
- 412 John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region
413 policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR,
414 2015.
- 415 Stephen Boyd, Lin Xiao, and Almir Mutapcic. Subgradient methods. *lecture notes of EE392o*,
416 *Stanford University, Autumn Quarter*, 2004:2004–2005, 2003.
- 417 Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In
418 *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 267–274,
419 2002.
- 420 Yongshuai Liu, Jiaxin Ding, and Xin Liu. Ipo: Interior-point policy optimization under constraints.
421 In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 4940–4947,
422 2020.
- 423 Tsung-Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J Ramadge. Projection-based
424 constrained policy optimization. *arXiv preprint arXiv:2010.03152*, 2020b.
- 425 Linrui Zhang, Qin Zhang, Li Shen, Bo Yuan, and Xueqian Wang. Saferl-kit: Evaluating efficient
426 reinforcement learning methods for safe autonomous driving. *arXiv preprint arXiv:2206.08528*,
427 2022b.
- 428 Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control.
429 In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033.
430 IEEE, 2012.