
Online Optimization for Offline Safe Reinforcement Learning

Yassine Chemingui

Washington State University
yassine.chemingui@wsu.edu

Aryan Deshwal

University of Minnesota
adeshwal@umn.edu

Alan Fern

Oregon State University
alan.fern@oregonstate.edu

Thanh Nguyen-Tang

New Jersey Institute of Technology
thanh.nguyen@njit.edu

Janardhan Rao Doppa

Washington State University
jana.doppa@wsu.edu

Abstract

We study the problem of Offline Safe Reinforcement Learning (OSRL), where the goal is to learn a reward-maximizing policy from fixed data under a cumulative cost constraint. We propose a novel OSRL approach that frames the problem as a minimax objective and solves it by combining offline RL with online optimization algorithms. We prove the approximate optimality of this approach when integrated with an approximate offline RL oracle and no-regret online optimization. We also present a practical approximation that can be combined with any offline RL algorithm, eliminating the need for offline policy evaluation. Empirical results on the DSRL benchmark demonstrate that our method reliably enforces safety constraints under stringent cost budgets, while achieving high rewards. The code is available at <https://github.com/yassineCh/OSRL>.

1 Introduction

Offline reinforcement learning (RL) [Levine et al., 2020] is a powerful paradigm to learn decision-making policies from logged datasets without the need for additional interaction with the environment. Offline RL has shown good success in domains including autonomous driving, robotics, control systems, and black-box optimization [Fang et al., 2022, Lin et al., 2024, Li et al., 2024, Zhan et al., 2022, Chemingui et al., 2024] where executing exploratory actions is not practical. However, in safety-critical domains such as healthcare and smart grid, the decision-making agent needs to also satisfy some cost/safety constraints. Such problems are studied under the sub-area referred to as offline safe RL (OSRL) where the goal is to learn reward-maximizing policies which satisfy cost constraints from offline datasets [Wachi et al., 2024].

OSRL inherits the challenges from both offline RL and safe RL [Wachi et al., 2024]. First, handling distributional shift when the learned policy encounters states and actions not seen in the offline dataset. Prior work in offline RL addresses this challenge by some form of penalization to realize the general principle of pessimism in the face of uncertainty [Levine et al., 2020]. Second, ensuring that the learned policy meets the cost/safety constraints after deployment. Maintaining the cost constraints typically requires off-policy evaluation (OPE) procedures which are highly unstable and typically have estimation errors [Figueiredo Prudencio et al., 2024]. OSRL is an active research topic and is relatively less-studied than offline RL. Constrained policy optimization using Lagrangian relaxation [Polosky et al., 2022, Lee et al., 2022, Xu et al., 2022b] is a popular approach for OSRL, but due to the need to solve intertwined optimization problems they are unstable in practice, and are shown to have bad performance either causing oscillation/divergence or learning overly conservative (near zero-reward) policies. Additionally, OSRL problem with stringent safety constraints (i.e., small cost

thresholds) is an important but severely under-studied problem [Zheng et al., 2024] and arises in many safety-critical control applications such as industrial process and energy system control.

This paper develops a novel offline constrained RL framework referred to as **Online Optimization for Offline Safe RL (O3SRL)**. O3SRL formulates a minimax optimization problem for policy learning and solves it using an iterative approach that relies on two key components, namely, an offline RL oracle and a no-regret algorithm. We perform two main algorithmic steps in each iteration of O3SRL. First, we produce a distribution over policies from solving an offline RL problem by defining a new reward function that combines the original reward and cost values based on the values of the current Lagrange variable and the cost threshold. Second, we employ a no-regret algorithm [Hazan et al., 2016] to adaptively update the value of Lagrange variable based on the current distribution over policies. We prove that the O3SRL framework is guaranteed to converge to the minimax solution.

Unfortunately, the general O3SRL framework suffers from two drawbacks in practice. First, the use of no-regret algorithms over a continuous Lagrange variable requires calling OPE procedures which are typically unstable [Figueiredo Prudencio et al., 2024] and can potentially lead to error propagation over iterations. Calling an OPE procedure in each iteration also contributes to computational expense. Second, it is computationally-expensive to run an offline RL algorithm to convergence in each iteration. To overcome these challenges, we develop a practical and effective algorithm with convergence guarantees. We employ K discrete values for Lagrange variable and apply a multi-armed bandit algorithm. Since most offline RL algorithms are based on stochastic gradient descent, we perform a small number of gradient updates on the policy from the previous iteration for efficiency. One advantage of this approach is that we can leverage powerful offline RL algorithms in a plug-and-play manner to solve offline constrained RL problems.

We perform experimental evaluation of this practical approach on multiple DSRL benchmark tasks [Liu et al., 2024] and our main findings are as follows. First, even the simplest version of our approach with two arms ($K=2$) is quite effective and results in state-of-the-art performance. Second, our approach achieves excellent performance for the challenging setting of small cost constraint thresholds. Third, the performance improves with higher arms ($K > 2$) but shows diminishing returns beyond $K=5$. Finally, O3SRL performs effectively with different offline RL algorithms and higher cost limits demonstrating its robust performance.

Contributions. The key contribution of this paper is the development and evaluation of the O3SRL framework to solve offline safe RL problems. Specific contributions include:

- Formulating OSRL as a minimax optimization problem and providing an iterative framework based on no-regret algorithms with convergence guarantees to the minimax solution.
- Development of a practical algorithm with convergence guarantee by avoiding the usage of off-policy evaluation (can lead to error propagation) and running offline RL algorithm to convergence (high computational cost) inside each iteration of a multi-round method.
- Empirical evaluation of the practical algorithm and its variants on DSRL benchmark tasks to demonstrate its effectiveness over state-of-the-art methods.

2 Problem Setup

RL problems with safety constraints are typically formulated using the Constrained Markov Decision Process (CMDP) framework [Altman, 2021]. A CMDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, P, r, c, \gamma, \mu)$, where \mathcal{S} and \mathcal{A} denote the state and action spaces respectively, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the unknown stochastic state transition function, $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function, $c : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}_{\geq 0}$ is the cost function, $\gamma \in (0, 1)$ is the discount factor, and μ is the initial state distribution.

Let $\pi : \mathcal{S} \rightarrow \mathcal{A}$ denote a policy and let $\tau = \{(s_t, a_t, r_t, c_t)\}_{t=1}^T$ be a trajectory of length T induced by π under dynamics P . The cumulative discounted return and cost for a trajectory τ are given by $R(\tau) = \sum_{t=1}^T \gamma^t r_t$ and $C(\tau) = \sum_{t=1}^T \gamma^t c_t$, respectively. In offline safe reinforcement learning (OSRL) problems, the agent is given access only to a static dataset $\mathcal{D}_{OSRL} = \{(s_i, a_i, r_i, c_i, s'_i)\}_{i=1}^n$ of n examples collected from an unknown behavior policy, and the overall goal is to learn a policy π to maximize the expected reward while satisfying a given constraint on the expected cumulative cost. Formally, the learning objective is as follows:

$$\max_{\pi} \mathbb{E}_{\tau \sim \pi} [R(\tau)] \quad \text{subject to} \quad \mathbb{E}_{\tau \sim \pi} [C(\tau)] \leq \kappa,$$

where $\kappa \geq 0$ is the maximum cost threshold/limit for safety constraint. This problem setting combines the challenges of offline RL (e.g., out-of-distribution states) with the added requirement of ensuring cost constraint satisfaction from static data alone, without online interaction.

Most of the prior work in OSRL cannot handle the setting where the cost threshold κ is small [Zheng et al., 2024] that is important for safety-critical applications with a tight budget. One of the key goals of this paper is to handle this challenging setting in a principled manner while solving the general OSRL problem effectively for higher cost constraint thresholds.

3 Related Work

Offline RL. This setting differs from the standard online RL setup in the following way: the goal is to learn reward maximizing policies from a given static dataset without interacting with the environment. The key technical challenge is to effectively handle out-of-distribution (OOD) states and actions [Levine et al., 2020, Figueiredo Prudencio et al., 2024]. Prior work has addressed this OOD challenge using improvements in value function estimation [Fujimoto and Gu, 2021, Kostrikov et al., 2022, Kumar et al., 2019, Lyu et al., 2022, Yang et al., 2022, Nguyen-Tang and Arora, 2023b,a], sequential modeling [Janner et al., 2021, Wang et al., 2022], uncertainty-awareness [An et al., 2021, Bai et al., 2022], constraining policy based on divergence [Wu et al., 2020, Jaques et al., 2020, Wu et al., 2022], by extending methods for model based RL [Kidambi et al., 2020, Yu et al., 2020, Rigter et al., 2022] and imitation learning [Xu et al., 2022a], and action selection based on filtering and reweighting [Chen et al., 2023, Hansen-Estruch et al., 2023].

Offline safe RL. This is a generalization of the offline RL setting where we need to learn a safe policy from a given offline dataset and a pre-specified cost constraint [Liu et al., 2024]. Offline safe RL is a relatively less-studied problem. Prior work includes methods based on constrained policy optimization [Polosky et al., 2022, Lee et al., 2022] and Lagrangian relaxation [Le et al., 2019, Xu et al., 2022b]. [Le et al., 2019] proposed one of the first batch-constrained frameworks that combines offline RL with online Lagrange-multiplier updates. However, this approach faces practical challenges (error propagation and high computational effort) due to the need to call an OPE procedure multiple times in each iteration which we address in this paper. We elaborate on these challenges and differences in the Appendix.

Extensions with convex MDP assumptions have also been explored [Zhang et al., 2024], along with approaches based on adversarial cost penalties [Wei et al., 2024], decision transformer [Liu et al., 2023], and diffusion models to create safe policies [Lin et al., 2023, Zheng et al., 2024, Yao et al., 2024]. CAPS [Chemingui et al., 2025] addresses dynamic safety constraints at deployment time by switching between a set of pre-trained policies based on the test-time cost budget. [Guo et al., 2025] introduced CCAC framework, which conditions the policy and value functions on cost thresholds to enable adaptation and generalization to unseen constraint levels. FISOR [Zheng et al., 2024] addresses safety by enforcing hard constraints—ensuring state-wise cost constraint satisfaction. Rather than optimizing toward a given cost limit, FISOR focuses on minimizing cost directly by learning policies that select actions only within the feasible region.

This paper aims to address the following two drawbacks of prior work: **1)** Lagrangian based constrained policy optimization methods solve inter-twined optimization problems and are typically unstable in practice resulting in poor performance; and **2)** While FISOR produces safe policies under tight cost constraints, it achieves low reward performance. The proposed O3SRL framework provides a theoretically-grounded and yet practically-effective solution that can be wrapped around existing offline RL methods and does not require an OPE procedure.

4 O3SRL Framework for Offline Constrained RL

In this section, we first provide a minimax optimization formulation of the offline constrained RL problem. Next, we describe the general O3SRL framework based on no-regret algorithms to solve this optimization problem along with theoretical guarantees for convergence to the minimax solution. Finally, we provide an approximate algorithm for practical purposes.

4.1 Minimax Optimization Formulation

The discounted value functions of a policy π for reward and cost objectives are defined as follows.

$$V_r^\pi := \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right] \quad \text{and} \quad V_c^\pi := \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \right]$$

where γ is the discount factor, $r(s_t, a_t)=r_t$ and $c(s_t, a_t)=c_t$ are the reward and cost values corresponding to state-action pair (s_t, a_t) respectively.

Let Π be the set of policies and $\Delta\Pi$ be the set of all distributions over Π . We aim to solve the following constrained optimization problem:

$$\max_{D \in \Delta\Pi} \mathbb{E}_{\pi \sim D} [V_r^\pi] \quad \text{subject to} \quad \mathbb{E}_{\pi \sim D} [V_c^\pi] \leq \kappa. \quad (1)$$

Let $L(D, \lambda)$ be the Lagrangian of Equation (1) where λ is the Lagrange multiplier.

$$L(D, \lambda) := \mathbb{E}_{\pi \sim D} [V_r^\pi] - \lambda(\mathbb{E}_{\pi \sim D} [V_c^\pi] - \kappa) = \mathbb{E}_{\pi \sim D} \left[V_r^\pi - \lambda(c - (1-\gamma)\kappa) \right]. \quad (2)$$

The dual problem of Equation (1) is given by:

$$\min_{\lambda \geq 0} \max_{D \in \Delta\Pi} L(D, \lambda). \quad (3)$$

Note that the objective and the constraint in Equation (1) are linear, thus convex and differentiable. Additionally, the constraint in Equation (1) satisfies Slater's condition when Π is expressive enough that there exists at least one policy $\pi \in \Pi$ that is strictly safe ($V_c^\pi \leq \kappa$), thus strong duality holds. This implies that solving the primal problem in Equation (1) is equivalent to solving the dual problem in Equation (3). It is also equivalent to constraining the domain of λ to any bounded domain $\Lambda = [0, C]$ for some $C > 0$: if (D^*, λ^*) is a solution to the dual problem and if $\lambda^* > 0$, then $(D^*, 1)$ is also a solution to the dual problem. This is because under strong duality, any solution (D^*, λ^*) of the dual must satisfies the KKT conditions: $\lambda(\mathbb{E}_{\pi \sim D} [V_c^\pi] - \kappa) = 0$.

Definition 1 (ϵ -approximate equilibrium). $(\hat{D}, \hat{\lambda}) \in \Delta\Pi \times \Lambda$ is ϵ -approximate minimax equilibrium of Equation (3) if

$$L(D, \hat{\lambda}) - \epsilon \leq L(\hat{D}, \hat{\lambda}) \leq L(\hat{D}, \lambda) + \epsilon, \forall (D, \lambda) \in \Delta\Pi \times \Lambda.$$

4.2 General O3SRL Framework via No-Regret Algorithms

Overview of the O3SRL Framework. O3SRL relies on two key elements: 1) an offline RL oracle which takes a training dataset of (state, action, next-state, and reward) tuples to produce a policy that maximizes reward, and 2) a no-regret algorithm which adaptively selects the value of Lagrange variable λ from a given search space Λ . We initialize λ with λ_0 and the distribution over policies D to D_0 respectively. In each iteration t , we perform two algorithmic steps. First, we employ the given offline safe RL training dataset $\mathcal{D}_{OSRL} = \{(s_i, a_i, r_i, c_i, s'_i)\}_{i=1}^n$ of n examples and convert it into a corresponding offline RL training dataset $\mathcal{D}_{ORL}^t = \{(s_i, a_i, r'_i, s'_i)\}_{i=1}^n$ by defining a new reward r' that combines original reward r and cost c values using λ_t based on the Equation 2: $r'_i = r_i - \lambda_{t-1}(c_i - (1-\gamma)\kappa)$ where γ is the discount factor and κ is the cost constraint threshold. We call an offline RL oracle on the dataset \mathcal{D}_{ORL}^t to get the distribution over policies D_t . Second, we call a no-regret algorithm [Hazan et al., 2016] by passing λ_{t-1} and D_t to get the updated Lagrange variable λ_t . At the end of T iterations, we return the average distribution of policies \bar{D} and the average Lagrangian value $\bar{\lambda}$ as the solution. Algorithm 1 provides a pseudo-code of the O3SRL framework.

In what follows, we first formally define offline RL oracle and no-regret algorithm. Next, we provide a theoretical result that guarantees that the O3SRL framework converges to the minimax solution.

Definition 2 (Offline RL oracle). Let $\mathbb{O}_{\text{offline-RL}}$ be an offline RL oracle that takes the offline data \mathcal{D}_f^n of size n from an MDP with the expected reward functions f (and state transition function P) and produces a policy distribution $D = \mathbb{O}_{\text{offline-RL}}(\mathcal{D}_f^n) \in \Delta\Pi$ at the estimation error $\epsilon_{\text{offline-RL}}(n)$, i.e.,

$$\mathbb{E}_{\mathcal{D}_f^n} \mathbb{E}_{\pi \sim \mathbb{O}_{\text{offline-RL}}(\mathcal{D}_f^n)} [V_f^* - V_f^\pi] \leq \epsilon_{\text{offline-RL}}(n). \quad (4)$$

Algorithm 1 General O3SRL Framework for Offline Constrained RL via No-Regret Algorithms

Require: Offline training data $\mathcal{D}_{OSRL} = \{(s_i, a_i, r_i, c_i, s'_i)\}_{i=1}^n$, Offline RL oracle $\mathbb{O}_{\text{offline-RL}}$, no-regret update oracle NO_REGRET_UPDATE , and search space of Lagrange variable Λ

1: Initialize $\lambda_0 \in \Lambda, D_0 \in \Delta\Pi$

2: **for** $t = 1, \dots, T$ **do**

3: $D_t \leftarrow \mathbb{O}_{\text{offline-RL}}(\mathcal{D}_{OSRL}^t = \{(s_i, a_i, r'_i = r_i - \lambda_{t-1}(c_i - (1 - \gamma)\kappa), s'_i)\}_{i=1}^n)$

4: $\lambda_t \leftarrow \text{NO_REGRET_UPDATE}(\lambda_{t-1}, D_t, \Lambda)$

5: **end for**

Ensure: $\bar{D} = \frac{1}{T} \sum_{t=1}^T D_t$ and $\bar{\lambda} = \frac{1}{T} \sum_{t=1}^T \lambda_t$ ▷ Parameter averaging

Remark 1. The estimation error of an offline RL algorithm $\epsilon_{\text{offline-RL}}(n)$ also depends on the coverage of the offline data for the optimal policy. This coverage in turn depends on the underlying MDP class, i.e., the stochastic transition function and reward function. In Algorithm 1, we employ an offline RL algorithm repeatedly for multiple MDPs with different reward functions. However, the same order rate $\epsilon_{\text{offline-RL}}(n)$ can be applied to multiple such offline RL guarantees, with perhaps at the cost of a constant factor. The main reason is that, most notions of coverage depend very mildly on the reward functions, if at all. For example, the concentrability coefficients [Liu et al., 2020, Rashidinejad et al., 2021] do not depend on the reward function at all. Similarly, the data diversity measure in [Nguyen-Tang and Arora, 2023a] and the policy transfer coefficient [Nguyen-Tang and Arora, 2024] do not depend on the specific reward function but rather a class of reward functions. In particular, if we use \mathcal{F} to approximate V_r^π and \mathcal{G} to approximate V_c^π , then $\mathcal{F} - \Lambda \cdot \mathcal{G} := \{f - \lambda \cdot g : f \in \mathcal{F}, g \in \mathcal{G}, \lambda \in \Lambda\}$ suffices to control the same policy transfer coefficient of a class of MDPs defined by any newly created reward functions used in Algorithm 1. Additionally, under good data coverage, we typically have $\epsilon_{\text{offline-RL}}(n) = \mathcal{O}(\frac{1}{\sqrt{n}})$.

Definition 3 (No-regret algorithm). *Let NO_REGRET_UPDATE is a no-regret algorithm for $\lambda \in \Lambda$ with regret bound $R_T(\Lambda)$, i.e.,*

$$\lambda_{t+1} = \text{NO_REGRET_UPDATE}(\lambda_t, D_t, \Lambda),$$

where

$$\sum_{t=1}^T L(D_t, \lambda_t) - \min_{\lambda \in \Lambda} \sum_{t=1}^T L(D_t, \lambda) \leq R_T(\Lambda) \quad (5)$$

and $\lim_{T \rightarrow \infty} \frac{R_T(\Lambda)}{T} \rightarrow 0$.

We consider a generic framework for solving offline constrained RL as shown in Algorithm 1. It relies on an offline RL oracle as per Definition 2 and a no-regret update oracle as per Definition 3.

Remark 2. From the given offline safe RL training dataset $\mathcal{D}_{OSRL} = \{(s_i, a_i, r_i, c_i, s'_i)\}_{i=1}^n$ of n examples, we can create a corresponding dataset that simulates the experiences of any MDP with the same unknown state transition probability function and an arbitrary reward function $r' = f(r, c)$, by simply computing $r'_i = f(r_i, c_i)$. Specifically, in our algorithm we define the modified reward function based on the Equation 2 as: $r' = r - \lambda_t(c - (1 - \gamma)\kappa)$ where $\lambda_t \in \Lambda$ is the Lagrange variable, γ is the discount factor, and κ is the cost constraint threshold.

The below theorem provides guarantee that Algorithm 1 will converge to the minimax solution.

Theorem 1. *Let $(\bar{D}, \bar{\lambda})$ be the output of Algorithm 1. Then, $(\bar{D}, \bar{\lambda})$ is ϵ -approximate equilibrium, where $\epsilon = \epsilon_{\text{offline-RL}}(n) + \frac{R_T(\Lambda)}{T}$.*

See the detailed proof in Appendix A.

5 Practical Algorithms

In this section, we introduce an approximation to Algorithm 1 that addresses two key practical challenges. Below we outline these challenges and the corresponding modifications. The resulting Algorithm 2 with minor approximation forms the basis of our primary empirical contribution.

The first practical challenge is that Algorithm 1 assumes access to an offline RL oracle, which is computationally expensive to approximate by running offline RL algorithms to convergence. This is especially problematic because the oracle must be invoked in every round of the algorithm. To mitigate this cost, we assume a weaker oracle: a *stochastic oracle* for offline RL in Assumption 4. This assumption says that we have access to an oracle that returns a near-optimal policy and a stochastic approximation of the value function of this near-optimal policy.

Definition 4. Let $\mathbb{S}\mathbb{O}_{\text{offline-RL}}$ be a stochastic oracle for offline RL that takes the offline data \mathcal{D}_f^n of size n from an MDP with the expected reward function f (and state transition function P across all MDP instances we have considered so far) and produces a policy distribution and a stochastic approximation of the value of the optimal policy: $(\tilde{D}, \tilde{V}) = \mathbb{S}\mathbb{O}_{\text{offline-RL}}(\mathcal{D}_f^n)$, where

$$\mathbb{E}_{\mathcal{D}_f^n} \mathbb{E}_{\pi \sim \tilde{D}} [V_f^* - V_f^\pi] \leq \epsilon_{\text{offline-RL}}(n), \text{ and } \mathbb{E}[\tilde{V}] = \mathbb{E}_{\pi \sim \tilde{D}} [V_f^\pi]. \quad (6)$$

This significantly relaxes the need to return an exact value of the value function of a near-optimal policy by an offline RL oracle, and, more importantly, leaves room for further practical approximation (as we elaborate in Section 5.1).

The second practical challenge arises from the use of no-regret algorithms over a continuous range of λ . The online optimization over a continuous domain of λ requires an off-policy estimate of a target policy under a mixed reward of the form $r - \lambda c$ for *every* value of λ . This often requires an OPE procedure (to compute $V_{c-(1-\gamma)\kappa}^{D^t}$ where D^t is the near-optimal policy distribution returned at line 4 in Algorithm 1) at every iteration. The application of OPE at every iteration is problematic in two ways: 1) Can compound the OPE estimation error exponentially over the length of the iterations, causing instability and large error; and 2) Employing OPE for every iteration is computationally expensive. To address this, we discretize the range of λ into K discrete values $\{\lambda^{(1)}, \dots, \lambda^{(K)}\}$ and optimize over this finite set. This transforms the problem into a multi-armed bandit (MAB) setting, where each of the K arms corresponds to a particular λ value. Crucially, many MAB algorithms do not require estimates of the underlying value functions, thus avoiding the need for OPE. In this work, we adopt the EXP3 algorithm [Auer et al., 2002a], which maintains a distribution over arms that is updated and sampled from in each round of Algorithm 2.

Algorithm 2 O3SRL Approximation via Stochastic Oracle and EXP3 Multi-Arm Bandit Strategy

Require: Stochastic oracle for offline RL $\mathbb{S}\mathbb{O}_{\text{offline-RL}}$, Offline safe RL training data $\mathcal{D}_{\text{OSRL}} =$

$\{(s_i, a_i, r_i, c_i, s'_i)\}_{i=1}^n$, and search space $\Lambda = \{\lambda^{(1)}, \dots, \lambda^{(K)}\}$

1: Initialize $P_0(\lambda) = \frac{1}{K}$, $\bar{c} = \kappa/H$, where κ is cost limit and $H = \frac{1}{1-\gamma}$ is the horizon

2: **for** $t = 1, \dots, T$ **do**

3: $(\tilde{D}_t, \tilde{V}_t) = \mathbb{S}\mathbb{O}_{\text{offline-RL}}(\mathcal{D}_{\text{ORL}}^t = \{(s_i, a_i, r'_i = r_i - \lambda_{t-1}(c_i - (1-\gamma)\kappa), s'_i)\}_{i=1}^n)$

4: Sample $\lambda_t \sim P_t$ where $P_t(\lambda) \propto P_{t-1}(\lambda) \exp\left(-\eta \frac{\tilde{V}_t \mathbb{1}\{\lambda = \lambda_{t-1}\}}{P_{t-1}(\lambda)}\right)$

5: **end for**

Ensure: $\bar{D} = \text{Proj}_{\Delta\Pi}\left(\frac{1}{T} \sum_{t=1}^T \tilde{D}_t\right)$, $\bar{\lambda} = \text{Proj}_{\Lambda}\left(\frac{1}{T} \sum_{t=1}^T \lambda_t\right)$

Theorem 2. Let $\Lambda = \{\lambda^{(1)}, \dots, \lambda^{(K)}\}$, where $\lambda^1 = 0$ and $\lambda^{(i+1)} - \lambda^{(i)} = \frac{C}{K}$. Then, the solution $(\bar{D}, \bar{\lambda})$ returned by Algorithm 2 is ϵ -approximate minimax equilibrium of Equation (3), where

$$\epsilon = \mathcal{O}\left(\underbrace{\epsilon_{\text{offline-RL}}(n)}_{\text{error rate of offline RL}} + \underbrace{\sqrt{\frac{K}{T}}}_{\text{regret of EXP3}} + \underbrace{\frac{1}{K}}_{\text{discretization error}} \right).$$

Remark 3: If we set $K = \Theta(T^{1/3})$ and assume the minimax rate $\frac{1}{\sqrt{n}}$ of offline RL algorithm, then the error rate in Theorem 2 will be in the order of $\frac{1}{\sqrt{n}} + \frac{1}{\sqrt[3]{T}}$. This rate is a bit slower than the rate $\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{T}}$ had we have applied a no-regret algorithm such as Follow-The-Regularized-Leader [Hazan et al., 2016] over continuous values of λ .

5.1 Final Approximate Algorithm for Empirical Evaluation

We consider an even more practical version of Algorithm 2 to form the final algorithm for our empirical evaluation. Specifically, we make two changes.

- We further approximate the stochastic oracle by leveraging the fact that most offline RL algorithms are based on stochastic gradient descent. In our approximation (Algorithm 2), rather than running the offline RL algorithm to convergence at round t , we perform only M gradient updates, after initializing from the result at the end of round $t - 1$. We find this truncated update scheme to be effective even for small values of M .
- Algorithm 2 (and Algorithm 1) maintains an averaged distribution over policies across iterations that is returned at the end of T rounds. In our case, this would involve storing the policy produced on each round, which would be prohibitively memory intensive. Instead, we simply return the last-iterate policy π_T .

In our experiments, we have found that this set of approximations leads to state-of-the-art performance even for as few as $K=2$ arms and a small number of Offline RL iterations M per round.

6 Experiments and Results

In this section, we experimentally evaluate the performance of the approximate O3SRL algorithm described in Section 5.1, compare with state-of-the-art baselines, and discuss the results.

6.1 Experimental Research Questions

Our experiments are designed to answer the following key questions:

Q1: What is the influence of the number of arms in our EXP-3 based approximate O3SRL approach on both reward and safety objectives?

Q2: How effectively does O3SRL satisfy safety/cost constraints and achieve high reward performance for small cost constraint thresholds?

Q3: How does the safety and reward trade-off of O3SRL vary as the cost threshold increases?

Q4: Is O3SRL compatible with different offline RL algorithms, demonstrating its generality?

6.2 Experimental Setup

Benchmarks. We evaluate O3SRL and baseline methods using the DSRL Bullet benchmark [Liu et al., 2024], which provides standardized offline datasets for safe RL research and evaluation. We consider eight continuous control tasks, comprising four “Run” and four “Circle” environments. Each task involves a different agent interacting with either the Run or Circle objective, following the common Agent-Task naming convention (e.g., Car-Run, Drone-Circle).

In the Run tasks, agents are rewarded for moving quickly between two boundaries while adhering to safety constraints, such as velocity limits and positional bounds. In the Circle tasks, agents are encouraged to follow a circular trajectory within a restricted safe region. These are challenging environments which require agents to balance performance with constraint satisfaction.

Evaluation metrics. Following the DSRL evaluation protocol [Liu et al., 2024], we evaluate all methods using normalized cumulative reward and normalized cumulative cost. Let $r_{\max}(M)$ and $r_{\min}(M)$ denote the maximum and minimum empirical reward returns achievable for task M . The normalized reward is computed as $R_{\text{normalized}} = \frac{R_{\pi} - r_{\min}(M)}{r_{\max}(M) - r_{\min}(M)}$, where R_{π} is the cumulative reward of policy π . The normalized cost is given by $C_{\text{normalized}} = \frac{C_{\pi}}{\kappa}$, with κ being the cost budget.

where C_{π} is the cumulative cost of the policy π and κ is the cost threshold. A policy π is considered safe if $C_{\text{normalized}} \leq 1$. Our O3SRL method is general-purpose and can be trained for any given cost threshold κ . However, we report our main results under a stringent low cost threshold of $\kappa=5$ to

highlight the method’s effectiveness in highly constrained settings. We report both mean and standard deviations over three random seeds, with each seed’s policy evaluated across twenty episodes.

Baseline methods. We compare O3SRL with several state-of-the-art offline safe RL methods. **(1)** BC-Safe is a behavior cloning baseline trained exclusively on safe trajectories that satisfy the given cost limit. **(2)** BEAR-Lag is a Lagrangian-based extension of BEAR [Kumar et al., 2019], which incorporates safety constraints during training. **(3)** CPQ [Xu et al., 2022b] penalizes unsafe actions by treating out-of-distribution actions as inherently risky and updates its Q-function using only safe transitions. **(4)** COptiDICE [Lee et al., 2022] builds on OptiDICE [Lee et al., 2021] by applying stationary distribution correction under cost constraints. **(5)** CDT [Liu et al., 2023] is a transformer-based method that conditions policies on returns and costs, allowing constraint-aware sequence modeling. **(6)** CCAC [Guo et al., 2025] learns a policy that adapts to changing safety budgets by jointly modeling constraints and environment dynamics. This approach conditions both the policy and value functions on constraint information. **(7)** CAPS [Chemingui et al., 2025], similar to CDT and CCAC, supports test-time adaptation to varying cost constraints by switching between a set of pre-trained policies with different cost and reward trade-offs. **(8)** FISOR [Zheng et al., 2024] is a diffusion-based method that explicitly optimizes for feasibility, aims to produce zero-violation policies, and is the state-of-the-art method for stringent cost constraints. All baselines are evaluated under the same cost constraint threshold $\kappa = 5$. All results are averaged over three random seeds, with each policy evaluated across 20 episodes. Additional details are in the Appendix.

Configuration of O3SRL. We implement the approximate O3SRL algorithm described in Section 5.1 using TD3+BC [Fujimoto and Gu, 2021] as the underlying offline RL algorithm for our main results. We set the search space of Lagrange variable $\Lambda = [0, C=5]$ and use $K=5$ arms for the main results but we also present ablation results for $K=2, 5$, and 10. We observe that increasing K beyond 5 (say to 10) yields only marginal improvements compared to the gains observed when increasing from $K=2$ to $K=5$. The algorithm is trained for $T = 100,000$ iterations, with arm probabilities updated every $M = 10$ steps of the base offline RL algorithm (i.e., $M=10$ stochastic gradient updates of the policy). For baselines, we employ the DSRL [Liu et al., 2024] implementation wherever available. For CCAC, CAPS and FISOR, we use the publicly released code provided by their respective authors. We employ IQL based instantiation of CAPS in our evaluation. Full architectural and hyperparameter details are provided in the Appendix.

6.3 Results and Discussion

Performance of O3SRL vs. the number of arms K . To evaluate the impact of the granularity of discretization of λ , we test O3SRL using $K \in \{2, 5, 10\}$ arms. Table 1 shows the corresponding results. Across all six benchmark tasks, O3SRL satisfies the cost constraint under every configuration and maintains consistently high reward, demonstrating that the EXP3 strategy is effective even with a coarse discretization of λ . With just two arms, the method keeps average cost at or near zero, but tends to underperform slightly in terms of rewards. Increasing to ten arms improves rewards marginally. The five-arm configuration achieves a good trade-off: it either matches or exceeds the highest feasible reward across all tasks while keeping costs safely below the threshold. For this reason, we adopted $K = 5$ configuration of the approximate O3SRL approach for all main results.

Table 1: O3SRL results for different values of K : the number of arms. Each value shows the average reward (\uparrow) and cost (\downarrow). Higher reward is better, and lower cost (up to threshold 1) is better.

# Arms	BallRun		CarRun		AntRun		BallCircle		CarCircle		AntCircle	
	reward \uparrow	cost \downarrow	reward \uparrow	cost \downarrow	reward \uparrow	cost \downarrow	reward \uparrow	cost \downarrow	reward \uparrow	cost \downarrow	reward \uparrow	cost \downarrow
$K = 2$	0.20	0.00	0.96	0.01	0.24	0.03	0.62	0.00	0.64	0.00	0.44	0.04
$K = 5$	0.25	0.00	0.96	0.02	0.33	0.14	0.62	0.06	0.66	0.11	0.48	0.00
$K = 10$	0.27	0.00	0.96	0.00	0.29	0.17	0.64	0.28	0.67	0.15	0.49	0.00

O3SRL vs. Baselines. Table 2 presents the comparative evaluation of O3SRL against baselines across eight DSRL tasks. The table reports both normalized reward (higher is better) and normalized cost (lower is better, with values less than 1 indicating constraint satisfaction). O3SRL is the *only* method that consistently satisfies the cost constraint across all eight tasks. This is particularly noteworthy given the stringent cost budget, which poses a significant challenge for all baseline algorithms.

Table 2: O3SRL normalized rewards and costs results. The cost threshold is 1. The \uparrow symbol denotes that higher rewards are better. The \downarrow symbol denotes that lower costs (up to threshold 1) are better. Each value is averaged over 20 evaluation episodes, and three random seeds. **Bold**: Safe agents whose normalized cost ≤ 1 . Gray: Unsafe agents. **Blue**: Safe agents with the highest reward.

Tasks		BC Safe	BEAR-Lag	CPQ	COptiDICE	CDT	CCAC	CAPS	FISOR	O3SRL
BallRun	Reward \uparrow	0.16 \pm 0.11	0.53 \pm 0.47	0.09 \pm 0.26	0.53 \pm 0.10	0.27 \pm 0.09	0.31\pm0.01	0.07\pm0.05	0.09 \pm 0.07	0.25\pm0.03
	Cost \downarrow	4.50 \pm 3.38	18.60 \pm 0.35	2.20 \pm 2.88	10.83 \pm 1.68	2.57 \pm 3.23	0.00\pm0.00	0.00\pm0.00	1.28 \pm 1.70	0.00\pm0.00
CarRun	Reward \uparrow	0.92\pm0.01	0.25 \pm 1.23	0.93\pm0.01	0.91\pm0.04	0.99\pm0.00	1.82 \pm 0.84	0.97\pm0.00	0.74\pm0.01	0.96\pm0.01
	Cost \downarrow	0.26\pm0.23	30.17 \pm 10.64	0.20\pm0.18	0.00\pm0.00	0.90\pm0.34	24.57 \pm 20.94	0.11\pm0.17	0.00\pm0.00	0.02\pm0.03
DroneRun	Reward \uparrow	0.41 \pm 0.23	-0.21 \pm 0.12	0.29 \pm 0.11	0.68 \pm 0.01	0.58\pm0.00	0.50 \pm 0.08	0.41 \pm 0.06	0.31 \pm 0.04	0.32\pm0.05
	Cost \downarrow	1.62 \pm 1.73	14.85 \pm 8.77	2.35 \pm 4.08	15.02 \pm 0.10	0.07\pm0.07	16.29 \pm 9.35	5.70 \pm 3.08	2.52 \pm 1.10	0.68\pm1.18
AntRun	Reward \uparrow	0.56 \pm 0.02	0.02\pm0.02	0.03\pm0.05	0.61 \pm 0.01	0.70 \pm 0.03	0.03\pm0.10	0.53 \pm 0.13	0.43\pm0.02	0.33\pm0.13
	Cost \downarrow	1.15 \pm 0.47	0.00\pm0.01	0.05\pm0.08	3.26 \pm 1.39	1.66 \pm 0.24	0.00\pm0.00	2.03 \pm 2.17	0.27\pm0.15	0.14\pm0.10
BallCircle	Reward \uparrow	0.45 \pm 0.03	0.85 \pm 0.05	0.56 \pm 0.10	0.71 \pm 0.01	0.61 \pm 0.17	0.47 \pm 0.38	0.33\pm0.02	0.32\pm0.05	0.62\pm0.01
	Cost \downarrow	1.21 \pm 0.23	10.92 \pm 2.02	1.11 \pm 1.92	9.41 \pm 0.69	2.03 \pm 1.39	10.19 \pm 17.65	0.01\pm0.02	0.00\pm0.00	0.06\pm0.07
CarCircle	Reward \uparrow	0.31 \pm 0.06	0.73 \pm 0.05	0.71\pm0.02	0.49 \pm 0.01	0.71 \pm 0.01	0.70 \pm 0.01	0.40\pm0.03	0.37\pm0.02	0.66\pm0.03
	Cost \downarrow	1.57 \pm 0.07	7.39 \pm 1.28	0.00\pm0.00	10.82 \pm 1.28	1.58 \pm 0.57	1.61 \pm 0.51	0.03\pm0.03	0.00\pm0.00	0.11\pm0.16
DroneCircle	Reward \uparrow	0.50 \pm 0.01	0.84 \pm 0.07	-0.21\pm0.04	0.26 \pm 0.02	0.55 \pm 0.01	0.43\pm0.04	0.36\pm0.02	0.48\pm0.01	0.49\pm0.07
	Cost \downarrow	1.22 \pm 0.31	15.40 \pm 2.62	0.70\pm0.41	3.68 \pm 0.51	1.14 \pm 0.06	0.06\pm0.07	0.00\pm0.00	0.17\pm0.15	0.23\pm0.34
AntCircle	Reward \uparrow	0.40 \pm 0.02	0.67 \pm 0.05	0.00\pm0.00	0.18 \pm 0.02	0.45 \pm 0.05	0.49\pm0.07	0.33\pm0.05	0.24\pm0.02	0.48\pm0.06
	Cost \downarrow	4.72 \pm 0.87	19.00 \pm 1.68	0.00\pm0.00	17.40 \pm 0.36	6.59 \pm 1.42	0.02\pm0.003	0.00\pm0.00	0.04\pm0.08	0.00\pm0.00

O3SRL’s consistent satisfaction of the cost constraint stands in contrast to methods that only occasionally achieve strong reward, often by violating safety. For instance, CDT achieves high rewards on tasks such as *CarRun* and *DroneRun*, but exceeds the safety budget in the other six tasks—making it unreliable. Similarly, CCAC achieves the best rewards on *BallRun* and *AntCircle*, yet violates the cost constraint in four out of eight tasks, further underscoring the challenge of maintaining both high performance and safety across environments. CPQ, on the other hand, stays within the budget more often, but its performance is inconsistent: it achieves a competitive reward on *CarCircle*, while falling short on rewards when it does remain safe, and it exceeds the cost threshold in three tasks.

While CAPS and FISOR maintain safe behavior in six out of eight tasks, their reward performance consistently trails behind O3SRL. For example, in *BallRun*, O3SRL is one of the only three safe methods (along with CAPS and CCAC), yet it achieves a reward of 0.25 ± 0.03 , which is over three times higher than CAPS (0.07 ± 0.05). A similar pattern holds in *BallCircle*, where O3SRL maintains safety and attains 0.65 ± 0.01 reward—more than double that of CAPS (0.23 ± 0.17) and FISOR (0.27 ± 0.15), both of which are also safe in that task. Note that FISOR, which aims for zero cost policies, fails to meet the cost threshold in the *DroneRun* and *BallRun* environments.

Overall, the results in Table 2 demonstrate that O3SRL strikes a uniquely favorable balance between maximizing rewards and satisfying safety constraints. It consistently ensures safety where many other methods fail, especially under a tight cost budget. Crucially, this safety does not come at a severe expense to the reward performance; O3SRL is the best-performing method (in terms of reward) among safe agents in two out of eight tasks, and consistently ranks second in the rest. This suggests that O3SRL’s iterative, no-regret approach to optimizing the reward and safety trade-off is more robust and effective than direct Lagrangian optimization or specialized heuristic-based safety mechanisms in these challenging offline safe RL settings.

Performance of O3SRL vs. increasing cost limits. To assess the flexibility of O3SRL in handling different safety constraints, we perform an ablation study across two extra cost limits beyond the default setting of $\kappa = 5$ used in the main experiments. Table 3 shows the corresponding results. We find that O3SRL continues to perform effectively under higher cost limits (i.e., less stringent safety constraints), adjusting its behavior to make better use of the available budget. As the constraint becomes more permissive, the learned policy shifts toward higher-reward strategies, while still respecting the specified cost limit. These results demonstrate that the O3SRL method does not rely on tuning for a particular budget.

O3SRL with an alternative Offline RL algorithm. To evaluate the generality of O3SRL with respect to the underlying offline RL algorithm, we conduct an ablation using Implicit Q-Learning (IQL) [Kostrikov et al., 2022] in place of TD3+BC, which is used in our main experiments, on the Circle tasks. Table 4 shows the corresponding results. We observe that O3SRL combined with

Table 3: O3SRL results for different values of κ : cost budget. Each value shows the average reward (\uparrow) and cost (\downarrow). Higher reward is better, and lower cost (up to threshold 1) is better.

Cost Limit	BallRun		DroneRun		BallCircle		DroneCircle	
	reward \uparrow	cost \downarrow	reward \uparrow	cost \downarrow	reward \uparrow	cost \downarrow	reward \uparrow	cost \downarrow
$\kappa = 5$	0.25	0.00	0.32	0.68	0.62	0.06	0.49	0.23
$\kappa = 20$	0.26	0.06	0.38	0.41	0.69	0.58	0.53	0.36
$\kappa = 40$	0.53	0.71	0.38	0.44	0.76	0.64	0.65	0.85

IQL achieves similar performance to the TD3+BC variant, both in satisfying safety constraints and attaining competitive reward. These results demonstrate that O3SRL is not tied to a specific offline RL method and can effectively incorporate different algorithms, highlighting its flexibility as a general framework for offline safe reinforcement learning.

Table 4: O3SRL results for different base offline RL methods. Each value shows the average reward (\uparrow) and cost (\downarrow). Higher reward is better, and lower cost (up to threshold 1) is better.

Base Offline RL	BallCircle		CarCircle		DroneCircle		AntCircle	
	reward \uparrow	cost \downarrow	reward \uparrow	cost \downarrow	reward \uparrow	cost \downarrow	reward \uparrow	cost \downarrow
TD3BC	0.62	0.06	0.66	0.11	0.49	0.23	0.48	0.00
IQL	0.54	0.06	0.63	0.10	0.42	0.03	0.41	0.06

In Appendix, we provide extended experimental results including evaluation on more benchmark tasks; treating λ as a fixed hyper-parameter; and ablation results for O3SRL over search space of λ and discretization levels, and the number of stochastic gradient updates of offline RL (M).

7 Summary and Future Work

This paper developed and evaluated a novel online optimization framework called O3SRL to create safe and reward-maximizing decision policies from offline datasets. O3SRL solves a minimax optimization problem by employing an iterative approach that adaptively updates the policy distribution using an offline RL oracle and Lagrange variables in each iteration. To handle the practical challenges, we studied an approximate algorithm that works over discrete values of Lagrange variables (to avoid offline policy evaluation) and performs few gradient updates of an offline RL algorithm (for computational efficiency). Empirical results show that our approximate O3SRL algorithm outperforms state-of-the-art methods, especially for stringent safety/cost constraints. Future work includes applying O3SRL to real-world applications and extending it to the offline-to-online safe RL setting.

Acknowledgments

The authors gratefully acknowledge the in part support by USDA-NIFA funded AgAID Institute award 2021-67021- 35344. The views expressed are those of the authors and do not reflect the official policy or position of the USDA-NIFA.

References

- E. Altman. *Constrained Markov decision processes*. Routledge, 2021.
- G. An, S. Moon, J.-H. Kim, and H. O. Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in neural information processing systems*, 34:7436–7447, 2021.
- P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002a. doi: 10.1137/S0097539701398375.
- P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002b.
- C. Bai, L. Wang, Z. Yang, Z.-H. Deng, A. Garg, P. Liu, and Z. Wang. Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=Y4cs1Z3HnqL>.
- Y. Chemingui, A. Deshwal, T. N. Hoang, and J. R. Doppa. Offline model-based optimization via policy-guided gradient search. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024*, pages 11230–11239. AAAI Press, 2024.
- Y. Chemingui, A. Deshwal, H. Wei, A. Fern, and J. Doppa. Constraint-adaptive policy switching for offline safe reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 15722–15730, 2025.
- H. Chen, C. Lu, C. Ying, H. Su, and J. Zhu. Offline reinforcement learning via high-fidelity generative behavior modeling. In *The Eleventh International Conference on Learning Representations*, 2023.
- X. Fang, Q. Zhang, Y. Gao, and D. Zhao. Offline reinforcement learning for autonomous driving with real world driving data. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 3417–3422. IEEE, 2022.
- R. Figueiredo Prudencio, M. R. O. A. Maximo, and E. L. Colombini. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE Transactions on Neural Networks and Learning Systems*, 35(8):10237–10257, 2024. doi: 10.1109/TNNLS.2023.3250269.
- S. Fujimoto and S. S. Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.
- Z. Guo, W. Zhou, S. Wang, and W. Li. Constraint-conditioned actor-critic for offline safe reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025.
- P. Hansen-Estruch, I. Kostrikov, M. Janner, J. G. Kuba, and S. Levine. Idql: Implicit q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*, 2023.
- E. Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.
- M. Janner, Q. Li, and S. Levine. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34:1273–1286, 2021.
- N. Jaques, A. Ghandeharioun, J. H. Shen, C. Ferguson, A. Lapedriza, N. Jones, S. Gu, and R. Picard. Way off-policy batch deep reinforcement learning of human preferences in dialog, 2020. URL <https://openreview.net/forum?id=rJl5rRVFvH>.
- R. Kidambi, A. Rajeswaran, P. Netrapalli, and T. Joachims. Morel: Model-based offline reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21810–21823. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/f7ef4f864ae9b88d43527f4b14f750f-Paper.pdf.
- I. Kostrikov, A. Nair, and S. Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=68n2s9ZJWF8>.

- A. Kumar, J. Fu, M. Soh, G. Tucker, and S. Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in neural information processing systems*, 32, 2019.
- H. Le, C. Voloshin, and Y. Yue. Batch policy learning under constraints. In *International Conference on Machine Learning*, pages 3703–3712. PMLR, 2019.
- J. Lee, W. Jeon, B. Lee, J. Pineau, and K.-E. Kim. Optidice: Offline policy optimization via stationary distribution correction estimation. In *International Conference on Machine Learning*, pages 6120–6130. PMLR, 2021.
- J. Lee, C. Paduraru, D. J. Mankowitz, N. Heess, D. Precup, K.-E. Kim, and A. Guez. COptiDICE: Offline constrained reinforcement learning via stationary distribution correction estimation. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=FLA55mBee6Q>.
- S. Levine, A. Kumar, G. Tucker, and J. Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.
- J. Li, X. Liu, B. Zhu, J. Jiao, M. Tomizuka, C. Tang, and W. Zhan. Guided online distillation: Promoting safe reinforcement learning by offline demonstration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7447–7454. IEEE, 2024.
- H. Lin, W. Ding, Z. Liu, Y. Niu, J. Zhu, Y. Niu, and D. Zhao. Safety-aware causal representation for trustworthy offline reinforcement learning in autonomous driving. *IEEE Robotics and Automation Letters*, 2024.
- Q. Lin, B. Tang, Z. Wu, C. Yu, S. Mao, Q. Xie, X. Wang, and D. Wang. Safe offline reinforcement learning with real-time budget constraints. In *International Conference on Machine Learning*, pages 21127–21152. PMLR, 2023.
- Y. Liu, A. Swaminathan, A. Agarwal, and E. Brunskill. Off-policy policy gradient with stationary distribution correction. In *Uncertainty in artificial intelligence*, pages 1180–1190. PMLR, 2020.
- Z. Liu, Z. Guo, Y. Yao, Z. Cen, W. Yu, T. Zhang, and D. Zhao. Constrained decision transformer for offline safe reinforcement learning. In *International Conference on Machine Learning*, pages 21611–21630. PMLR, 2023.
- Z. Liu, Z. Guo, H. Lin, Y. Yao, J. Zhu, Z. Cen, H. Hu, W. Yu, T. Zhang, J. Tan, and D. Zhao. Datasets and benchmarks for offline safe reinforcement learning. *Journal of Data-centric Machine Learning Research*, 2024.
- J. Lyu, X. Ma, X. Li, and Z. Lu. Mildly conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35:1711–1724, 2022.
- T. Nguyen-Tang and R. Arora. On sample-efficient offline reinforcement learning: Data diversity, posterior sampling and beyond. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL <https://openreview.net/forum?id=sd1h4gV0j8>.
- T. Nguyen-Tang and R. Arora. Viper: Provably efficient algorithm for offline rl with neural function approximation. *arXiv preprint arXiv:2302.12780*, 2023b.
- T. Nguyen-Tang and R. Arora. On the statistical complexity of offline decision-making. In *Proceedings of the 41st International Conference on Machine Learning*, pages 37900–37928, 2024.
- N. Polosky, B. C. Da Silva, M. Fiterau, and J. Jagannath. Constrained offline policy optimization. In *International Conference on Machine Learning*, pages 17801–17810. PMLR, 2022.
- P. Rashidinejad, B. Zhu, C. Ma, J. Jiao, and S. Russell. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information Processing Systems*, 34: 11702–11716, 2021.
- M. Rigter, B. Lacerda, and N. Hawes. Rambo-rl: Robust adversarial model-based offline reinforcement learning. *Advances in neural information processing systems*, 35:16082–16097, 2022.

- A. Wachi, X. Shen, and Y. Sui. A survey of constraint formulations in safe reinforcement learning. *arXiv preprint arXiv:2402.02025*, 2024.
- K. Wang, H. Zhao, X. Luo, K. Ren, W. Zhang, and D. Li. Bootstrapped transformer for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35:34748–34761, 2022.
- H. Wei, X. Peng, A. Ghosh, and X. Liu. Adversarially trained weighted actor-critic for safe offline reinforcement learning. *Advances in Neural Information Processing Systems*, 37:52806–52835, 2024.
- J. Wu, H. Wu, Z. Qiu, J. Wang, and M. Long. Supported policy optimization for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35:31278–31291, 2022.
- Y. Wu, G. Tucker, and O. Nachum. Behavior regularized offline reinforcement learning, 2020. URL <https://openreview.net/forum?id=BJg9hTNKPH>.
- H. Xu, L. Jiang, L. Jianxiong, and X. Zhan. A policy-guided imitation approach for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35:4085–4098, 2022a.
- H. Xu, X. Zhan, and X. Zhu. Constraints penalized q-learning for safe offline reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8753–8760, 2022b.
- R. Yang, C. Bai, X. Ma, Z. Wang, C. Zhang, and L. Han. RORL: Robust offline reinforcement learning via conservative smoothing. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=_QzJJGH_KE.
- Y. Yao, Z. Cen, W. Ding, H. Lin, S. Liu, T. Zhang, W. Yu, and D. Zhao. Oasis: Conditional distribution shaping for offline safe reinforcement learning. *Advances in Neural Information Processing Systems*, 37:78451–78478, 2024.
- T. Yu, G. Thomas, L. Yu, S. Ermon, J. Y. Zou, S. Levine, C. Finn, and T. Ma. Mopo: Model-based offline policy optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 14129–14142. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/a322852ce0df73e204b7e67cbbef0d0a-Paper.pdf.
- X. Zhan, H. Xu, Y. Zhang, X. Zhu, H. Yin, and Y. Zheng. Deepthermal: Combustion optimization for thermal power generating units using offline reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4680–4688, 2022.
- H. Zhang, X. Peng, H. Wei, and X. Liu. Safe and efficient: A primal-dual method for offline convex mdps under partial data coverage. *Advances in Neural Information Processing Systems*, 37:34239–34269, 2024.
- Y. Zheng, J. Li, D. Yu, Y. Yang, S. E. Li, X. Zhan, and J. Liu. Safe offline reinforcement learning with feasibility-guided diffusion model. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=j5JvZCaDM0>.

A Proofs for Theorem 1 and Theorem 2

Proof of Theorem 1. For any $D \in \Delta\Pi$, we have

$$\begin{aligned}
L(D, \bar{\lambda}) &= \frac{1}{T} \sum_{t=1}^T L(D, \lambda_t) && (L(D, \lambda) \text{ is linear in } \lambda) \\
&\leq \frac{1}{T} \sum_{t=1}^T (L(D_t, \lambda_t) + \epsilon_{\text{offline-RL}}(n)) && (\text{due to Equation (4)}) \\
&\leq \frac{1}{T} \sum_{t=1}^T L(D_t, \bar{\lambda}) + \epsilon_{\text{offline-RL}}(n) + \frac{R_T(\Lambda)}{T} && (\text{due to Equation (5)}) \\
&= L(\bar{D}, \bar{\lambda}) + \epsilon_{\text{offline-RL}}(n) + \frac{R_T(\Lambda)}{T} && (L(D, \lambda) \text{ is linear in } D).
\end{aligned}$$

For any $\lambda \in \Lambda$, we have

$$\begin{aligned}
L(\bar{D}, \lambda) &= \frac{1}{T} \sum_{t=1}^T L(D_t, \lambda) \\
&\geq \frac{1}{T} \sum_{t=1}^T L(D_t, \lambda_t) - \frac{R_T(\Lambda)}{T} && (\text{due to Equation (5)}) \\
&\geq \frac{1}{T} \sum_{t=1}^T L(\bar{D}, \lambda_t) - \epsilon_{\text{offline-RL}}(n) - \frac{R_T(\Lambda)}{T} && (\text{due to Equation (4)}) \\
&= L(\bar{D}, \bar{\lambda}) - \epsilon_{\text{offline-RL}}(n) - \frac{R_T(\Lambda)}{T}.
\end{aligned}$$

□

Proof of Theorem 2. The result directly follows from the assumption of the stochastic oracle in Definition 4, the regret bound of EXP3 [Auer et al., 2002b], and the approximation error for the projection from $[0, C]$ into the discrete domain Λ with resolution $1/K$ is $1/K$.

In particular, by [Auer et al., 2002b], we have

$$\sum_{t=1}^T L(\tilde{D}_t, \lambda_t) - \min_{\lambda \in \Lambda} \sum_{t=1}^T L(\tilde{D}_t, \lambda) \leq H\sqrt{2}\sqrt{TK \ln K}.$$

Applying the result in Theorem 1, we have

$$\max_{D \in \Delta\Pi} L(D, \hat{\lambda}) + \epsilon_1 \leq L(\bar{D}, \hat{\lambda}) \leq \min_{\lambda \in \Lambda} L(\bar{D}, \lambda) + \epsilon_1,$$

where

$$\epsilon_1 = \epsilon_{\text{offline-RL}}(n) + H\sqrt{\frac{2K \log K}{T}}, \text{ and } \hat{\lambda} := \frac{1}{T} \sum_{t=1}^T \lambda_t.$$

Note that the running average $\hat{\lambda}$ does not guarantee to be in Λ . Thus, with $\bar{\lambda} = \text{Proj}_{\Lambda}(\hat{\lambda})$ – a projection of $\hat{\lambda}$ onto Λ , we have that,

$$|L(D, \hat{\lambda}) - L(D, \bar{\lambda})| \leq \frac{H}{K}.$$

Overall, we have

$$\max_{D \in \Delta\Pi} L(D, \bar{\lambda}) + \epsilon_1 + \frac{H}{K} \leq L(\bar{D}, \bar{\lambda}) \leq \min_{\lambda \in \Lambda} L(\bar{D}, \lambda) + \epsilon_1 + \frac{H}{K},$$

Thus, $(\bar{D}, \bar{\lambda})$ is ϵ -approximate equilibrium, where

$$\epsilon = \epsilon_{\text{offline-RL}}(n) + H\sqrt{\frac{2K \log K}{T}} + \frac{H}{K}.$$

□

B Extended Discussion of Related Work

Lagrangian formulation is quite natural for offline safe RL that both our work and [Le et al., 2019] share. [Le et al., 2019] and our work also share the general idea that we can use online optimization algorithms to tune the Lagrangian multiplier. The critical difference is in the design of efficient and effective algorithmic frameworks: our framework requires *one call to an offline RL oracle per loop* and *does not rely on any OPE oracle*. On the other hand, [Le et al., 2019] requires *two calls to an offline RL oracle and four calls to an OPE oracle per loop*: four calls to OPE oracle in each iteration can lead to error propagation and exponential error over the number of iterations posing stability challenges; and six calls to oracles of different nature in each iteration leads to huge computational expense posing scalability challenges. Our approximate O3SRL algorithmic framework overcomes these challenges in a principled manner.

C Ablation Results for O3SRL

We provide additional ablation results for O3SRL beyond those presented in the main paper.

C.1 Extended Evaluation of O3SRL

To further evaluate the generalization capability of O3SRL, we extended our experiments to additional environments from the *Gymnasium Safety Suite*, following the FISOR [Zheng et al., 2024] setup with a tight cost limit of 10. The results are summarized in Table 5.

Table 5: O3SRL normalized rewards and costs results across SafetyGym environments. \uparrow denotes that higher rewards are better; \downarrow denotes that lower costs (up to threshold 1) are better. **Bold**: Safe agents whose normalized cost ≤ 1 . Gray: Unsafe agents. **Blue**: Safe agents with the highest reward.

Environment		CAPS	FISOR	CCAC	O3SRL
PointCircle1	Reward \uparrow	0.31\pm0.05	0.43 \pm 0.05	0.57 \pm 0.08	0.53 \pm 0.02
	Cost \downarrow	0.94\pm1.53	14.93 \pm 4.24	6.65 \pm 3.51	1.31 \pm 0.62
PointCircle2	Reward \uparrow	0.44\pm0.06	0.76 \pm 0.05	0.03 \pm 0.75	0.52\pm0.04
	Cost \downarrow	0.10\pm0.09	18.02 \pm 4.17	3.99 \pm 4.30	0.55\pm0.41
AntVelocity	Reward \uparrow	0.87\pm0.04	0.89\pm0.01	-1.01\pm0.00	0.45\pm0.43
	Cost \downarrow	0.19\pm0.06	0.00\pm0.00	0.00\pm0.00	0.05\pm0.06
HalfCheetah	Reward \uparrow	0.86\pm0.01	0.89\pm0.01	0.91\pm0.04	0.93\pm0.02
	Cost \downarrow	0.27\pm0.07	0.00\pm0.00	0.97\pm0.12	0.00\pm0.00
HopperVelocity	Reward \uparrow	0.29 \pm 0.14	0.12\pm0.03	-0.01\pm0.02	0.51\pm0.31
	Cost \downarrow	1.61 \pm 1.53	0.90\pm1.07	0.00\pm0.00	0.20\pm0.35
SwimmerVelocity	Reward \uparrow	0.38 \pm 0.10	-0.02\pm0.05	0.06\pm0.25	0.05\pm0.02
	Cost \downarrow	2.11 \pm 3.01	0.23\pm0.20	0.91\pm1.57	0.00\pm0.01
Walker2dVelocity	Reward \uparrow	0.78\pm0.01	0.51 \pm 0.03	0.40 \pm 0.17	0.77\pm0.02
	Cost \downarrow	0.08\pm0.07	1.92 \pm 0.57	5.20 \pm 1.16	0.02\pm0.02

O3SRL satisfies the cost constraint in 6 out of the 7 evaluated tasks, demonstrating strong generalization across environments with varying dynamics and cost structures. The next best-performing method, CAPS, remains safe on 5 out of 7 tasks. In terms of reward, O3SRL achieves the highest normalized return on three environments and remains competitive on the remaining tasks. For example, O3SRL attains performance close to the best methods on *SwimmerVelocity* (0.05 vs. 0.06) and *Walker2dVelocity* (0.77 vs. 0.78) while maintaining safety.

Collectively, these results show that O3SRL maintains safety under strict cost constraints while achieving competitive or superior performance across diverse safety-critical environments.

C.2 Alternative Approach: Offline RL with Fixed λ as a Hyperparameter

To assess the effect of treating the Lagrange multiplier λ as a fixed hyperparameter, we conducted an additional experiment in which λ was held constant throughout training, and the resulting policy

was evaluated at the end. This setup departs from the intended offline RL formulation, as choosing an appropriate λ would typically require access to online validation data or environment rollouts. Moreover, an exhaustive search over possible λ values is computationally impractical and inconsistent with the offline setting.

In our proposed approach (O3SRL), we instead define a discrete set of candidate λ values, each corresponding to an ‘‘arm’’ in a multi-armed bandit framework. During training, a λ value is sampled from a learned distribution over these arms and used to shape the reward. The sampling distribution is updated at every batch following the EXP3 algorithm, enabling adaptive reweighting based on the observed performance of each arm. This mechanism allows the policy to dynamically adjust the effective penalty strength, balancing constraint satisfaction and reward maximization throughout training rather than committing to a single fixed λ .

Empirically, as shown in Table 6 fixed- λ policies tend to either ignore the cost constraints when λ is too small or over-penalize and sacrifice reward when λ is too large. By contrast, O3SRL adapts its choice of λ during offline training, achieving a more favorable trade-off between reward and cost across different constraint budgets.

Table 6: DroneCircle results with fixed arms under varying cost limits. The \uparrow symbol denotes that higher rewards are better; \downarrow denotes that lower costs (up to threshold 1) are better. **Bold**: Safe agents whose normalized cost ≤ 1 . Gray: Unsafe agents. **Blue**: Safe agents with the highest reward.

Cost Limit		Arm 0	Arm 1	Arm 2	Arm 3	Arm 4	O3SRL
5	Reward \uparrow	0.90	0.86	0.55	0.47	0.39	0.49
	Cost \downarrow	18.18	16.52	2.47	0.00	0.00	0.23
20	Reward \uparrow	0.90	0.87	0.75	0.49	0.40	0.53
	Cost \downarrow	4.54	4.33	2.80	0.04	0.00	0.36
40	Reward \uparrow	0.90	0.87	0.82	0.52	0.42	0.65
	Cost \downarrow	2.27	2.21	1.88	0.08	0.00	0.85

C.3 Search space of Lagrange variable λ

Recall that our search space for λ is $[0, C]$. Table 7 presents an ablation study on the impact of varying C , the maximum allowable value for the Lagrange variable λ , which determines how strongly constraint violations are penalized in O3SRL. Results are shown for $C = 2, 5$, and 10.

The ablation results highlight that small values of C (e.g., $C = 2$) often lead to insufficient constraint enforcement, resulting in catastrophic safety violations in some environments. For example, in *CarCircle* and *DroneCircle*, the average cost exceeds the threshold by a factor of over 8 and 13 respectively, completely defeating the purpose of safe learning. These cases show that when λ is capped too low, the algorithm lacks the leverage to penalize unsafe behavior adequately.

In contrast, increasing C to 5 or 10 significantly improves safety across all tasks. For instance, in *DroneCircle*, the cost drops from 13.32 at $C = 2$ to 0.23 at $C = 5$, and further to 0.00 at $C = 10$. However, in tasks with already low costs (e.g., *AntCircle*), large C values can lead to reduced reward, due to over-penalization.

Overall, $C = 5$ emerges as a robust choice, offering strong safety guarantees while maintaining high reward across environments. This ablation underscores the necessity of a sufficiently expressive λ range to enable O3SRL to satisfy constraints in practice.

C.4 Discretization of λ

Table 8 presents an ablation comparing two λ set choices used in O3SRL: a uniformly spaced set (e.g., $[0, 1.25, 2.5, 3.75, 5]$) and an adaptive set that retains the extreme values 0 and 5 but distributes intermediate values more densely near lower penalty regions (e.g., $[0, 0.5, 1.12, 2.5, 5]$). The purpose of the adaptive grid is to modulate constraint penalization based on the cost threshold κ , allowing the algorithm to apply smaller penalties in more permissive cost budget scenarios. The results show that while both grids perform similarly under strict cost threshold constraints ($\kappa = 5$), the adaptive grid provides clear advantages as the constraint becomes more relaxed. For instance, under $\kappa = 40$, the

Table 7: O3SRL results for different values of C : λ maximum value. Each value shows the average reward (\uparrow) and cost (\downarrow). Higher reward is better, and lower cost (up to threshold 1) is better. **Bold**: Safe agents whose normalized cost ≤ 1 . Gray: Unsafe agents.

Tasks		$C = 2$	$C = 5$	$C = 10$
BallRun	Reward \uparrow	0.28 \pm 0.01	0.25 \pm 0.03	0.25 \pm 0.01
	Cost \downarrow	0.00 \pm 0.00	0.00 \pm 0.00	0.00 \pm 0.00
CarRun	Reward \uparrow	0.97 \pm 0.00	0.96 \pm 0.01	0.95 \pm 0.01
	Cost \downarrow	0.06 \pm 0.10	0.02 \pm 0.03	0.00 \pm 0.00
DroneRun	Reward \uparrow	0.43 \pm 0.10	0.32 \pm 0.05	0.33 \pm 0.05
	Cost \downarrow	1.93 \pm 1.90	0.68 \pm 1.18	0.38 \pm 0.66
AntRun	Reward \uparrow	0.34 \pm 0.09	0.33 \pm 0.13	0.20 \pm 0.06
	Cost \downarrow	1.16 \pm 0.95	0.14 \pm 0.10	0.27 \pm 0.47
BallCircle	Reward \uparrow	0.82 \pm 0.06	0.62 \pm 0.01	0.58 \pm 0.03
	Cost \downarrow	7.20 \pm 2.99	0.06 \pm 0.07	0.01 \pm 0.01
CarCircle	Reward \uparrow	0.75 \pm 0.11	0.66 \pm 0.03	0.57 \pm 0.08
	Cost \downarrow	8.07 \pm 7.53	0.11 \pm 0.16	0.00 \pm 0.00
DroneCircle	Reward \uparrow	0.79 \pm 0.06	0.49 \pm 0.07	0.37 \pm 0.03
	Cost \downarrow	13.32 \pm 2.62	0.23 \pm 0.34	0.00 \pm 0.00
AntCircle	Reward \uparrow	0.58 \pm 0.02	0.48 \pm 0.06	0.41 \pm 0.02
	Cost \downarrow	0.25 \pm 0.26	0.00 \pm 0.00	0.04 \pm 0.07

adaptive grid allows significantly higher reward in tasks such as *DroneCircle* (reward improves from 0.45 to 0.65), by leveraging a smoother trade-off between reward maximization and cost constraint satisfaction. These findings suggest that finer resolution at low penalty levels enables better policy selection for higher cost constraint thresholds, making the adaptive grid more effective for balancing performance and safety.

Table 8: O3SRL results for different values of the set of λ s and cost limits κ . Each value shows the average reward (\uparrow) and cost (\downarrow). Higher reward is better, and lower cost (up to threshold 1) is better. **Bold**: Safe agents whose normalized cost ≤ 1 . Gray: Unsafe agents.

Tasks		$\kappa = 5$		$\kappa = 20$		$\kappa = 40$	
		uniform	adaptive	uniform	adaptive	uniform	adaptive
BallRun	Reward \uparrow	0.25 \pm 0.01	0.25 \pm 0.03	0.25 \pm 0.01	0.26 \pm 0.01	0.25 \pm 0.01	0.53 \pm 0.48
	Cost \downarrow	0.00 \pm 0.00	0.00 \pm 0.00	0.00 \pm 0.00	0.06 \pm 0.11	0.00 \pm 0.00	0.71 \pm 1.24
DroneRun	Reward \uparrow	0.39 \pm 0.04	0.32 \pm 0.05	0.31 \pm 0.11	0.38 \pm 0.05	0.30 \pm 0.05	0.38 \pm 0.10
	Cost \downarrow	1.95 \pm 1.97	0.68 \pm 1.18	0.22 \pm 0.33	0.41 \pm 0.41	0.60 \pm 0.72	0.44 \pm 0.44
BallCircle	Reward \uparrow	0.63 \pm 0.04	0.62 \pm 0.01	0.63 \pm 0.08	0.69 \pm 0.09	0.62 \pm 0.06	0.76 \pm 0.04
	Cost \downarrow	0.07 \pm 0.12	0.06 \pm 0.07	0.20 \pm 0.30	0.58 \pm 0.66	0.10 \pm 0.10	0.64 \pm 0.18
DroneCircle	Reward \uparrow	0.46 \pm 0.02	0.49 \pm 0.07	0.46 \pm 0.01	0.53 \pm 0.01	0.45 \pm 0.03	0.65 \pm 0.11
	Cost \downarrow	0.00 \pm 0.00	0.23 \pm 0.34	0.00 \pm 0.00	0.36 \pm 0.27	0.00 \pm 0.00	0.85 \pm 0.55

C.5 Number of update steps M

Recall that our approximate algorithm for empirical evaluation performs M stochastic gradient updates of a given offline RL algorithm in each iteration. Table 9 presents an ablation on the number of offline RL update steps M . This parameter controls how long the policy is trained before updating the λ value to balance reward and constraint satisfaction. The results demonstrate that $M = 10$ provides a stable trade-off between learning progress and responsiveness to constraint violations.

With $M = 1$, the frequent updates to λ overall increases costs adjusting the policy aggressively. Conversely, setting $M = 100$ delays λ updates, which can result in policies drifting away from feasible regions, as seen in the sharp increase in costs for DroneRun and DroneCircle. Overall, the ablation results validate that safe and effective offline learning in O3SRL requires coordinated progress on both the policy and the constraint via well-timed λ updates. The choice of $M = 10$ strikes this balance, supporting its use for all the results presented in the main paper.

Table 9: O3SRL results for different values of M : Offline RL update steps. Each value shows the average reward (\uparrow) and cost (\downarrow). Higher reward is better, and lower cost (up to threshold 1) is better. **Bold**: Safe agents whose normalized cost ≤ 1 . Gray: Unsafe agents.

Tasks		$M = 1$	$M = 10$	$M = 100$
BallRun	Reward \uparrow	0.25\pm0.03	0.25\pm0.03	0.26\pm0.03
	Cost \downarrow	0.37\pm0.64	0.00\pm0.00	0.00\pm0.00
CarRun	Reward \uparrow	0.96\pm0.00	0.96\pm0.01	0.96\pm0.00
	Cost \downarrow	0.00\pm0.01	0.02\pm0.03	0.01\pm0.01
DroneRun	Reward \uparrow	0.34 \pm 0.06	0.32\pm0.05	0.27 \pm 0.08
	Cost \downarrow	1.10 \pm 0.90	0.68\pm1.18	3.72 \pm 4.44
AntRun	Reward \uparrow	0.34\pm0.05	0.33\pm0.13	0.39\pm0.16
	Cost \downarrow	0.20\pm0.21	0.14\pm0.10	0.97\pm1.36
BallCircle	Reward \uparrow	0.67\pm0.00	0.62\pm0.01	0.67\pm0.05
	Cost \downarrow	0.25\pm0.19	0.06\pm0.07	0.61\pm0.88
CarCircle	Reward \uparrow	0.66\pm0.01	0.66\pm0.03	0.68\pm0.01
	Cost \downarrow	0.10\pm0.17	0.11\pm0.16	0.17\pm0.29
DroneCircle	Reward \uparrow	0.44\pm0.06	0.49\pm0.07	0.58 \pm 0.20
	Cost \downarrow	0.11\pm0.10	0.23\pm0.34	4.87 \pm 7.77
AntCircle	Reward \uparrow	0.41\pm0.06	0.48\pm0.06	0.49\pm0.01
	Cost \downarrow	0.02\pm0.03	0.00\pm0.00	0.00\pm0.00

C.6 Evaluation Across Multiple Random Seeds

To assess the stability of O3SRL with respect to random initialization, we extend our evaluation to include additional random seeds. Specifically, we compare results averaged over the original three seeds (10, 20, and 30) with those averaged over all six seeds (10, 20, 30, 40, 50, and 60). Table 10 reports the mean and standard deviation of rewards and costs under both settings.

The results across confirm the stability of O3SRL, showing consistent rewards and safe cost levels across all evaluated environments. The performance trends observed with three seeds largely persist when extended to six, indicating robust and reliable behavior under different random seeds.

D Experimental Details

This section provides additional details about the experimental setup and hyper-parameters.

D.1 Description of environments

We evaluate O3SRL performance in a set of environments from the **Bullet-Safety-Gym** suite. Each environment is defined by a combination of *robot type* and *task*, resulting in scenarios such as *BallRun*, *CarCircle*, *DroneRun*, and so on. The agent types include **Ball**, **Car**, **Drone**, and **Ant**, and each is evaluated under either the **Run** or **Circle** task. Figure 1 illustrates the Bullet-Safety-Gym environments, demonstrating various agent types (Ball, Car, Drone, Ant) performing the Run and Circle tasks.

Table 10: Comparison between 3-seed and 6-seed experiments. Reward (\uparrow) indicates higher is better, and Cost (\downarrow) indicates lower is better. **Bold**: Safe agents whose normalized cost ≤ 1 .

Environment		3 seeds	6 seeds
BallRun	Reward \uparrow	0.25\pm0.03	0.26\pm0.02
	Cost \downarrow	0.00\pm0.00	0.00\pm0.00
CarRun	Reward \uparrow	0.96\pm0.01	0.96\pm0.01
	Cost \downarrow	0.02\pm0.03	0.00\pm0.00
DroneRun	Reward \uparrow	0.32\pm0.05	0.37\pm0.03
	Cost \downarrow	0.68\pm1.18	0.74\pm0.67
AntRun	Reward \uparrow	0.33\pm0.13	0.22\pm0.10
	Cost \downarrow	0.14\pm0.10	0.17\pm0.19
BallCircle	Reward \uparrow	0.62\pm0.01	0.63\pm0.04
	Cost \downarrow	0.06\pm0.07	0.26\pm0.25
CarCircle	Reward \uparrow	0.66\pm0.03	0.66\pm0.03
	Cost \downarrow	0.11\pm0.16	0.04\pm0.10
DroneCircle	Reward \uparrow	0.49\pm0.07	0.47\pm0.03
	Cost \downarrow	0.23\pm0.34	0.02\pm0.04
AntCircle	Reward \uparrow	0.48\pm0.06	0.44\pm0.05
	Cost \downarrow	0.00\pm0.00	0.02\pm0.03

Run Environments: In the Run task, agents are required to move quickly along a straight corridor. They receive rewards based on forward progress, encouraging high-speed locomotion. However, they are penalized for crossing lateral safety boundaries or exceeding a predefined velocity threshold. These safety constraints are not enforced through physical barriers but through a cost signal that tracks violations.

Circle Environments: In the Circle task, agents are trained to move in a clockwise direction around a circular track. Rewards are higher when agents maintain fast, smooth motion near the boundary of the circle. Deviating from the intended path or exiting the safety zone results in penalties.

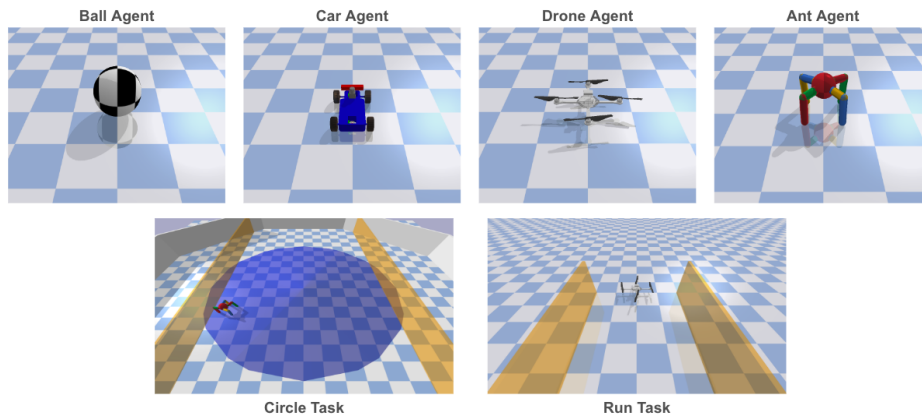


Figure 1: Visualization of the Bullet-Safety-Gym environments featuring different agents and tasks.

D.2 Training details

Our implementation is available at <https://anonymous.4open.science/r/O3SRL-E55F/>.

Adaptive λ set: We construct the λ set using a geometric spacing scheme that adapts to the cost limit. The grid always includes $\lambda = 0$ and a maximum value $\lambda_{\max} = C$, and places $k - 2$ intermediate points between them. These intermediate values are spaced geometrically and scaled by a factor

$\text{shrink} = (\text{reference_limit}/\text{cost_limit})^{\alpha_{\text{shrink}}}$, which compresses the grid as the cost limit increases, where $\text{reference_limit} = 5$ and $\alpha_{\text{shrink}} = 0.3$. This design ensures finer resolution among low-penalty values when constraints are loose, and broader coverage when strict cost constraints require stronger penalization.

Rewards scale: We observe that reward distributions in offline datasets can be heavy-tailed, with occasional extreme values that disproportionately affect learning stability, as illustrated in Figure 2.

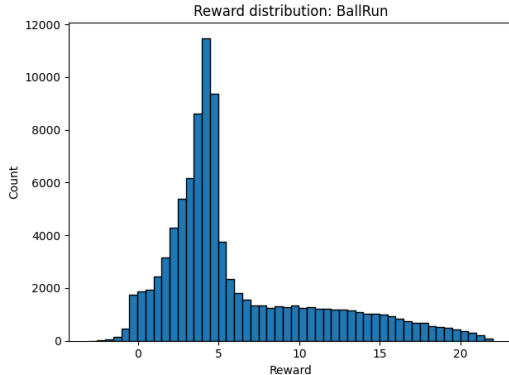


Figure 2: An example of a heavy-tailed reward distribution from the *BallRun* dataset.

To mitigate this, we clip rewards symmetrically at the τ -th percentile of their absolute values:

$$r_{\text{clip}} = \text{percentile}(|r|, \tau), \quad r \leftarrow \text{clip}(r, -r_{\text{clip}}, r_{\text{clip}})$$

where $\tau = 99$. We then scale down the clipped rewards multiplying by $0.9/r_{\text{clip}}$.

Hyperparameters: We employ a batch size of 1024 for the tasks *CarCircle*, *DroneRun*, *AntCircle*, and *BallCircle*, while a batch size of 512 is used for the remaining tasks: *CarRun*, *DroneCircle*, *AntRun*, and *BallRun*. The other hyperparameters used in our experiments are listed in Table 11. For TD3+BC, we adopt the same hyperparameter settings as in the original paper.

Table 11: O3SRL Hyperparameters

EXP3 parameters	
Number of arms k	5
Maximum C	5
λ update frequency M	10
λ learning rate η	2e-3
TD3BC parameters	
Discount γ	0.99
Policy noise	0.2
Policy noise clip	(0.5, 0.5)
Policy update frequency	2
α	2.5
Optimizer	Adam
Actor, Critic learning rate	3e-4
Actor, Critic hidden size	256
Training steps	100000
Seed	[10, 20, 30]

Training Curves of O3SRL : Figure 3 shows the training curves of O3SRL across the different tasks, plotting both the average reward and constraint cost over training iterations. The curves

demonstrate that O3SRL is able to quickly stabilize and achieve a favorable trade-off between reward maximization and cost minimization. In most tasks, the cost decreases steadily and remains below the constraint threshold, while the reward improves progressively.

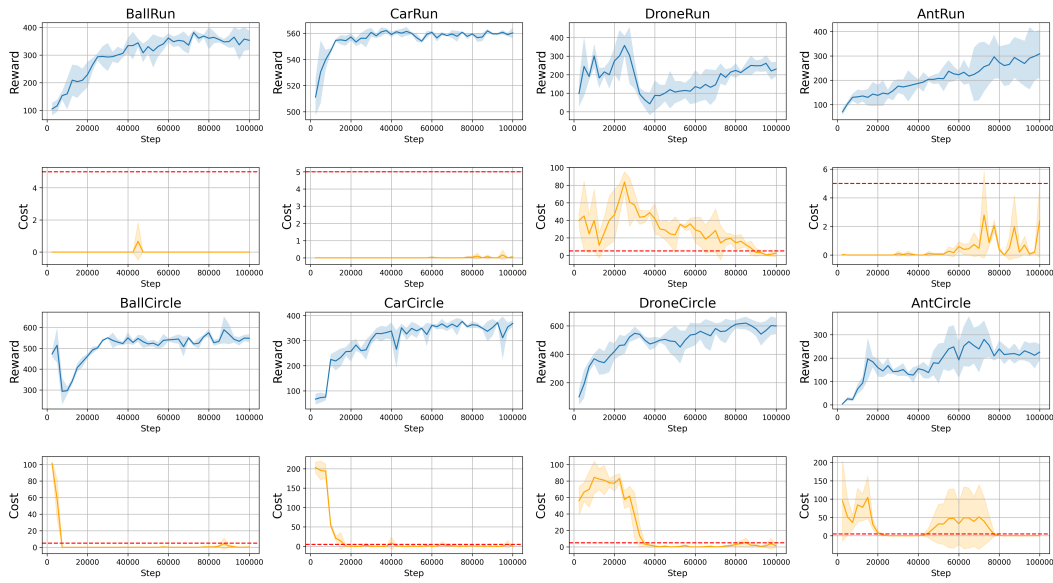


Figure 3: Training curves of O3SRL across different tasks, showing average reward (top) and cost (bottom) over training iterations.

Computational Resources and Training Time: All experiments were conducted using an NVIDIA A40 GPU with 48GB of memory. Training a single task for one random seed takes approximately 20 minutes when the server is not under load.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [NA].
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes]" is generally preferable to "[No]", it is perfectly acceptable to answer "[No]" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No]" or "[NA]" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist",**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers.**

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: The abstract and introduction claim to propose a practical OSRL framework based on a min-max game solved with no-regret algorithms, which reliably enforces safety constraints while maintaining strong reward performance. The paper develops this framework (Algorithm 1 and 2), provides theoretical convergence guarantees (Theorem 1 and 2), and empirically demonstrates its effectiveness on the DSRL benchmark (Table 1), aligning with these claims.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: The paper explicitly discusses several limitations of its proposed method, particularly in Section 5. It identifies two practical challenges: (1) the instability and computational cost associated with using off-policy evaluation (OPE) procedures and continuous no-regret optimization over the Lagrange multiplier, and (2) the inefficiency of running offline RL algorithms to convergence during each iteration. To address these, the paper introduces a practical approximation using discrete values for the Lagrange variable and truncated gradient updates. These discussions acknowledge trade-offs in approximation quality, computational cost, and potential performance limitations, satisfying the guideline for reflecting on practical constraints and assumptions of the approach.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The paper presents two main theoretical results: Theorem 1 concerning the convergence of the general framework (Algorithm 1) and Theorem 2 for the practical algorithm (Algorithm 2). Assumptions for these theorems (e.g., access to an offline RL oracle with a defined error, properties of the no-regret algorithm, discretization of λ) are stated. A proof sketch for Theorem 1 is provided in Appendix A.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper details the experimental setup in Section 6.1, including the benchmark used (DSRL Bullet benchmark), specific tasks, evaluation metrics (normalized reward and cost with formulas), and the cost budget ($\kappa=5$). It lists the configuration of O3SRL and all baseline methods. The full stepwise pseudocode of O3SRL (Algorithm 2) is described, including the use of EXP3. This level of detail should allow for reproduction of the main experimental comparisons.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer:[Yes]

Justification: Our code and data will be made available on a public GitHub repository. We also provide the anonymous code repository as supplementary for reviewing purposes.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper provides sufficient detail to understand and reproduce the training and test setup. In Section 6.2 ("Experimental Setup"), it specifies the benchmark used (DSRL Bullet), describes the tasks, and outlines the evaluation metrics including how normalized rewards and costs are computed. It mentions key parameters such as the cost threshold ($= 5$), the number of evaluation episodes (20), the number of training iterations ($T = 100,000$), the number of gradient steps per round ($M = 10$), and the number of random seeds (3). Additionally, the base offline RL algorithm (TD3+BC) is specified. For further transparency, the paper states that additional architectural and hyperparameter details are provided in the Appendix

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Table 1 reports mean values along with standard deviation for both normalized reward and cost evaluated over three random seeds which allows for an assessment of variability and statistical significance.

Guidelines:

- The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Details on the computational resources used — including hardware specifications (GPU/CPU type, memory), and approximate training time — are provided in the appendix to support reproducibility.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines?>

Answer: [Yes]

Justification: The research fully conforms to the NeurIPS Code of ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: The research focuses on methodological development for offline safe reinforcement learning, and we do not see any direct societal impacts of the work that would require specific discussion.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All third-party assets used in this paper are properly credited and used in accordance with their licenses. The DSRL Bullet benchmark datasets and associated implementation are released under the Apache License 2.0, which permits use and modification with attribution. For the CAPS and FISOR baseline methods, we use the publicly available implementations provided by the original authors on GitHub. We cite all relevant works in the paper and comply with their usage terms as outlined in their respective repositories

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The primary new asset introduced in this paper is the O3SRL algorithm. The paper provide its implementation in the appendix, along with sufficient documentation to reproduce the main results. This includes code and configuration details needed to run the experiments.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The research does not use LLMs as part of the core method development or contributions.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.