ZAPD: ZERO-SHOT ADAPTATION VIA PRIMAL-DUAL UPDATES FOR CONSTRAINED REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

In constrained reinforcement learning (CRL), agents maximize expected reward while ensuring the expected cost does not exceed a given threshold. Most existing CRL methods, including the widely used primal-dual approaches, assume this threshold is fixed, but in real-world applications it can shift abruptly. Standard CRL methods then need substantial re-learning, and existing threshold-adaptive methods require access to a range of thresholds or a library of pre-trained policies. To address this problem, we propose **Z**ero-shot **A**daptation via **P**rimal-**D**ual updates (ZAPD), which adapts the primal-dual solution to a new threshold without any additional environment interaction. ZAPD treats the shift as a perturbation of the converged solution and employs a first-order sensitivity analysis to derive closed-form primal and dual updates. Both updates are computed from data the underlying method already collects, so ZAPD plugs into a broad range of primal-dual CRL algorithms. Experiments show that ZAPD achieves the lowest immediate approximation error on all four Bullet Safety Gym tasks in the single-threshold setting, and that this improved initialization translates into faster post-shift convergence than threshold-adaptive baselines.

1 INTRODUCTION

When applying Reinforcement Learning (RL) (Sutton & Barto, 1998) to real-world problems, we are often faced with settings in which an agent must not only maximize task performance but also satisfy constraints. For example, a robot may aim to reach a goal location while avoiding collisions or limiting energy consumption (Huang et al., 2024; Ray et al., 2019; Ji et al., 2023). To model such problems, it is necessary to extend the standard RL framework to explicitly account for constraints. Constrained Reinforcement Learning (CRL) addresses this need by solving Constrained Markov Decision Processes (CMDPs) (Altman, 2021; Garcıa & Fernández, 2015), which provide a standard framework for optimizing expected reward subject to expected cost constraints.

Most CRL methods solve a CMDP for a threshold that is fixed in advance. A common approach is the Lagrangian formulation, which introduces a multiplier for each constraint and yields a min-max problem solved by primal-dual updates (Tessler et al., 2019; Ding et al., 2020; Stooke et al., 2020; Paternain et al., 2019; Borkar, 2005), as in PPO-Lagrangian (Schulman et al., 2017; Ji et al., 2024). Other methods enforce the constraint without an explicit multiplier, such as CPO (Achiam et al., 2017) and IPO (Liu et al., 2020). Recent primal-dual methods (Montenegro et al., 2024; Ding et al., 2023) have established global last-iterate convergence guarantees, but under fixed constraint thresholds. In practice, thresholds change: a safety envelope tightens as a robot approaches a person, an energy budget relaxes once conditions improve, or a regulatory limit is revised. Because the converged solution is tied to the threshold it was trained under, such a change forces the agent to re-learn through additional environment interaction. Neither direction is benign. When a constraint tightens, the stale policy violates it throughout re-learning; when it loosens, the stale policy forgoes reward it is now permitted to collect. Thus, faster reconvergence shortens the period of constraint violation or forgone reward during deployment.

Yet the pre-shift solution is not an arbitrary starting point: it solves a problem that differs from the new one only in the threshold. This observation motivates the following question:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Can an agent adapt immediately to an abrupt constraint threshold shift, using only its pre-shift converged primal–dual solution?

We answer this question affirmatively with Zero-shot Adaptation via Primal-Dual updates (ZAPD). Our key idea is to treat a threshold shift not as a new problem to be solved, but as a perturbation of a problem already solved. The converged solution satisfies a stationarity condition that depends on the threshold, so the sensitivity of that condition determines how the solution moves when the threshold moves. A first-order sensitivity analysis makes this precise and yields closed-form corrections to both the policy parameters and the multipliers. The corrections require only a constraint gradient and a curvature estimate, both computable from data the agent has already collected, so ZAPD adapts at the moment of the shift without any further environment interaction.

Our approach differs from existing threshold-adaptive CRL methods. CCPO (Yao et al., 2023) and CAPS (Chemingui et al., 2025) achieve adaptivity by training over a range of thresholds or by maintaining a library of policies pre-trained at several thresholds. Such prior information may be unavailable, and it is unclear how densely it must cover the relevant range without knowing the task-specific constraint boundaries in advance. OMDPD (Zhu et al., 2025) adapts to varying cost functions but assumes finite state and action spaces, which excludes the continuous-control settings we consider. Additional related approaches are provided in Appendix A. More fundamentally, these methods treat different thresholds as different optimization problems, to be solved separately and then selected among or interpolated between. We instead exploit the fact that the threshold itself has changed, which relates the pre-shift and post-shift solutions directly and requires no prior coverage of the threshold range. Section 5 shows that this yields lower immediate approximation error and faster reconvergence than both methods.

Contributions

1. **Problem Formulation.** We formalize the problem of abrupt constraint threshold shift in CRL (Section 2).
2. **First-Order Sensitivity Analysis.** We quantify how the converged primal-dual solution moves when the constraint threshold shifts, and obtain closed-form expressions for both the primal and the dual correction (Section 3).
3. **ZAPD Algorithm.** We turn these expressions into a practical algorithm that estimates the required quantities from pre-shift data, and that attaches to an underlying CRL method as a plug-in module (Section 4).
4. **Empirical Results.** We show that ZAPD achieves the lowest immediate approximation error on all four Bullet Safety Gym tasks against CCPO and CAPS, and that the gains transfer across base methods and policy parameterizations as a plug-in to C-PG and PPO-Lagrangian (Section 5).

2 PRELIMINARIES

In this section, we give a brief overview of the definition of CMDPs and the primal-dual CRL framework, and then formalize the abrupt constraint threshold shift setting.

We consider a discounted CMDP (Altman, 2021) $(\mathcal{S}, \mathcal{A}, P, r, \boldsymbol{c}, \gamma, \mu)$ with m constraints, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, P is the transition kernel, r is the reward function, $\boldsymbol{c} = (c_1, \ldots, c_m)^\top$ collects the m cost functions, $\gamma \in [0, 1)$ is the discount factor, and μ is the initial state distribution. We consider a policy parameterized by θ, denoted π_θ. The expected discounted reward under π_θ is $\mathcal{R}(\theta) := \mathbb{E}_{\pi_\theta}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$, and the expected discounted costs are collected in the vector $\boldsymbol{\mathcal{J}}(\theta) := (\mathcal{J}_1(\theta), \ldots, \mathcal{J}_m(\theta))^\top \in \mathbb{R}^m$, where $\mathcal{J}_i(\theta) := \mathbb{E}_{\pi_\theta}[\sum_{t=0}^{\infty} \gamma^t c_i(s_t, a_t)]$. Given a vector of cost thresholds $\boldsymbol{b} \in \mathbb{R}^m$, the CMDP objective is to maximize the expected discounted reward while ensuring that no expected discounted cost exceeds its threshold, that is:

$$\max_{\theta} \ \mathcal{R}(\theta) \quad \text{subject to} \quad \boldsymbol{\mathcal{J}}(\theta) \preceq \boldsymbol{b}, \tag{1}$$

where $\preceq$ denotes componentwise inequality, so that Eq. (1) requires $\mathcal{J}_i(\theta) \leq b_i$ for every $i \in \{1, \ldots, m\}$. The single-constraint setting is recovered by taking $m = 1$.

A standard approach to solving Eq. (1) is Lagrangian relaxation (Bertsekas, 1999), which introduces a multiplier $\boldsymbol{\lambda} \in \mathbb{R}^m$, one per constraint, to form the dual problem:

$$\min_{\boldsymbol{\lambda}\succeq\mathbf{0}} \max_{\theta} \mathcal{L}(\theta,\boldsymbol{\lambda}), \quad \text{where} \quad \mathcal{L}(\theta,\boldsymbol{\lambda}) := \mathcal{R}(\theta) - \boldsymbol{\lambda}^\top\big(\mathcal{J}(\theta) - \boldsymbol{b}\big). \tag{2}$$

The min-max structure of the dual problem naturally suggests a primal-dual method, where the algorithm iteratively performs a primal update on θ and a dual update on $\boldsymbol{\lambda}$, based on the current iterate $(\theta,\boldsymbol{\lambda})$ and data collected under the current policy. Concretely, at iteration k, the updates take the general form:

$$\theta_k = f(\theta_{k-1},\boldsymbol{\lambda}_{k-1};\boldsymbol{b}), \quad \boldsymbol{\lambda}_k = g(\theta_{k-1},\boldsymbol{\lambda}_{k-1};\boldsymbol{b}), \tag{3}$$

where f and g denote the primal and dual update functions of the underlying method. For example, in gradient-based methods (Ding et al., 2020; Paternain et al., 2019; Montenegro et al., 2024), these variables are typically updated by taking gradient steps on $\mathcal{L}$, ascending in θ and descending in $\boldsymbol{\lambda}$, with the multiplier projected onto the feasible set $\{\boldsymbol{\lambda}\succeq\mathbf{0}\}$. Other methods depart from this template while retaining the same two-step iterative structure: CVPO (Liu et al., 2022) recasts the problem as probabilistic inference, alternating an E-step that solves analytically for an optimal feasible variational distribution with an M-step that fits π_θ by supervised learning; PCPO (Yang et al., 2020) takes a trust-region reward-improvement step and then projects the policy back onto the constraint set. However, these algorithms treat $\boldsymbol{b}$ as fixed: having converged to a solution $(\theta^*,\boldsymbol{\lambda}^*)$ under $\boldsymbol{b}$, they can only re-run their iterative updates when a threshold changes, which may require many iterations to reconverge. In the interim, the policy remains tuned to the old thresholds, resulting in suboptimal reward and/or substantial constraint violation.

We consider a single abrupt shift of the constraint thresholds at iteration k_c of the primal-dual updates, where $\boldsymbol{b}$ is replaced by a new threshold vector $\hat{\boldsymbol{b}}$ that is revealed exactly at k_c:

$$\boldsymbol{b}_k = \begin{cases} \boldsymbol{b}, & k < k_c, \\ \hat{\boldsymbol{b}}, & k \geq k_c, \end{cases} \qquad \hat{\boldsymbol{b}} = \boldsymbol{b} + \Delta\boldsymbol{b}. \tag{4}$$

Here, $\Delta\boldsymbol{b} \neq \mathbf{0}$ denotes the signed shift, and $\boldsymbol{b}_k$ is the threshold vector in force at iteration k. Individual components of $\Delta\boldsymbol{b}$ may be positive, negative, or zero, so a shift may tighten some constraints, loosen others, and leave the rest unchanged; we require only that at least one component is nonzero. Given the solution $(\theta^*,\boldsymbol{\lambda}^*)$ reached under $\boldsymbol{b}$, our goal is to produce adapted variables $(\tilde{\theta},\tilde{\boldsymbol{\lambda}})$ that approximate the corresponding solution $(\hat{\theta}^*,\hat{\boldsymbol{\lambda}}^*)$ under $\hat{\boldsymbol{b}}$, so that training resumes from a near-optimal point and reconverges faster. The adaptation must be zero-shot: it may use only pre-shift information and the observed $\hat{\boldsymbol{b}}$, without any further environment interaction. Fig. 1 summarizes this setting.

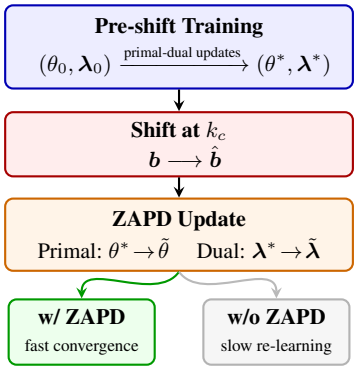

Figure 1: Overview of the abrupt constraint threshold shift setting.

3 Sensitivity Analysis for Abrupt Constraint Threshold Shifts

In this section, we use first-order sensitivity analysis to relate the pre-shift solution $(\theta^*,\boldsymbol{\lambda}^*)$ under $\boldsymbol{b}$ to the post-shift solution $(\hat{\theta}^*,\hat{\boldsymbol{\lambda}}^*)$ under $\hat{\boldsymbol{b}}$. In Section 4, we will use this relation to derive the primal and dual updates of ZAPD.

3.1 Assumptions

We first introduce some assumptions that we make throughout this section.

Assumption 1. *The pre-shift and post-shift solutions $(\theta^*,\boldsymbol{\lambda}^*)$ and $(\hat{\theta}^*,\hat{\boldsymbol{\lambda}}^*)$ are fixed points of the primal-dual updates in Eq. (3), so that $\nabla_\theta\mathcal{L}(\theta^*,\boldsymbol{\lambda}^*) = 0$ and $\nabla_\theta\mathcal{L}(\hat{\theta}^*,\hat{\boldsymbol{\lambda}}^*) = 0$.*

Assumption 2. *$\mathcal{R}(\theta)$ and $\mathcal{J}_i(\theta)$, $i = 1,\dots,m$, are twice differentiable, and the Hessian of $\mathcal{L}$, $H(\theta,\boldsymbol{\lambda}) := \nabla^2_\theta\mathcal{R}(\theta) - \sum_{i=1}^m \lambda_i\nabla^2_\theta\mathcal{J}_i(\theta)$, is invertible.*

Assumption 3. *For $\boldsymbol{\lambda}$ around $\boldsymbol{\lambda}^*$, let $\theta(\boldsymbol{\lambda}) := \arg\max_\theta \mathcal{L}(\theta,\boldsymbol{\lambda})$. The mapping $\boldsymbol{\lambda} \mapsto \theta(\boldsymbol{\lambda})$ is continuously differentiable, with Jacobian $\nabla_{\boldsymbol{\lambda}}\theta(\boldsymbol{\lambda}) \in \mathbb{R}^{n\times m}$, where n is the dimension of θ.*

Assumption 4. *All constraints are active at both solutions, i.e.,* $\boldsymbol{\mathcal{J}}(\theta^*) = \boldsymbol{b}$ *and* $\boldsymbol{\mathcal{J}}(\hat{\theta}^*) = \hat{\boldsymbol{b}}$.

Assumption 5. *The constraint Jacobian* $\nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*) \in \mathbb{R}^{m \times n}$ *has full row rank.*

Assumptions 1–3 are standard regularity conditions in the analysis of policy-gradient methods. Assumptions 4 and 5 together ensure that no constraint is redundant at θ^*, and this pre-shift requirement is not restrictive. To see why, suppose that either some constraint j is slack at θ^*, so that $\mathcal{J}_j(\theta^*) < b_j$, or that $\nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*)$ is rank deficient, so that some $\nabla_\theta \mathcal{J}_j(\theta^*)$ is a linear combination of the remaining constraint gradients. In either case constraint j imposes no first-order restriction at θ^* beyond those already imposed by the others: by Assumption 1, the stationarity condition $\nabla_\theta \mathcal{R}(\theta^*) = \sum_{i=1}^m \lambda_i^* \nabla_\theta \mathcal{J}_i(\theta^*)$ continues to hold once constraint j is dropped and the remaining multipliers are adjusted accordingly. We therefore remove such constraints, apply the analysis below to the reduced constraint set, and leave the multipliers of the removed constraints unchanged by the zero-shot update. Both conditions can be checked from pre-shift quantities. The remaining condition $\boldsymbol{\mathcal{J}}(\hat{\theta}^*) = \hat{\boldsymbol{b}}$ cannot be verified before the shift.

3.2 Quantifying the Primal Shift

By Assumption 4, we have $\boldsymbol{\mathcal{J}}(\theta^*) = \boldsymbol{b}$ and $\boldsymbol{\mathcal{J}}(\hat{\theta}^*) = \hat{\boldsymbol{b}}$. Subtracting these two equalities and expanding $\boldsymbol{\mathcal{J}}(\hat{\theta}^*)$ around θ^* yields:

$$\Delta \boldsymbol{b} \;=\; \hat{\boldsymbol{b}} - \boldsymbol{b} \;=\; \boldsymbol{\mathcal{J}}(\hat{\theta}^*) - \boldsymbol{\mathcal{J}}(\theta^*) \;=\; \nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*)(\hat{\theta}^* - \theta^*) \;+\; O\Big(\|\hat{\theta}^* - \theta^*\|_2^2\Big). \tag{5}$$

Here and below, a vector-valued $O(\cdot)$ term denotes a vector whose Euclidean norm is of the stated order. Dropping the remainder in Eq. (5) gives the linearized condition $\nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*)(\theta - \theta^*) = \Delta \boldsymbol{b}$. This is a system of m equations in the n unknowns θ, with m typically far smaller than n, so θ is underdetermined and we have no first-order information favoring any of the remaining directions. Since the expansion in Eq. (5) is accurate only near θ^*, we resolve this by taking the smallest such step, which results in the following optimization problem:

$$\min_\theta \; \frac{1}{2}\|\theta - \theta^*\|_2^2 \quad \text{subject to} \quad \nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*)(\theta - \theta^*) = \Delta \boldsymbol{b}. \tag{6}$$

Theorem 1 below shows that this constrained optimization problem has a unique closed-form solution, which we use as the primal shift approximation.

Theorem 1 (Primal shift approximation)**.** *Under Assumption 5, Eq. (6) has a unique solution* $\tilde{\theta}$*, given by*

$$\tilde{\theta} = \theta^* + \nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*)^\top \big(\nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*)\, \nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*)^\top\big)^{-1} \Delta \boldsymbol{b}. \tag{7}$$

When $m = 1$*, this reduces to*

$$\tilde{\theta} = \theta^* + \frac{\nabla_\theta \mathcal{J}(\theta^*)}{\|\nabla_\theta \mathcal{J}(\theta^*)\|_2^2} \Delta b.$$

The proof is provided in Appendix B.1. Eq. (7) involves only θ^*, the constraint Jacobian $\nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*)$, and the observed $\Delta \boldsymbol{b}$. It requires neither the multiplier $\boldsymbol{\lambda}^*$ nor any Hessian information, so the primal shift can be applied to any CRL method that produces a converged policy, including those that maintain no explicit multiplier.

3.3 Quantifying the Dual Shift

By the definition of $\theta(\boldsymbol{\lambda})$ in Assumption 3, $\nabla_\theta \mathcal{L}(\theta(\boldsymbol{\lambda}), \boldsymbol{\lambda}) = 0$, and Assumption 2 makes this stationary point locally unique. Hence, $\hat{\theta}^* = \theta(\hat{\boldsymbol{\lambda}}^*)$, $\theta^* = \theta(\boldsymbol{\lambda}^*)$. Applying a first-order Taylor expansion of $\theta(\boldsymbol{\lambda})$ around $\boldsymbol{\lambda}^*$ gives

$$\hat{\theta}^* - \theta^* = \nabla_{\boldsymbol{\lambda}} \theta(\boldsymbol{\lambda}^*)(\hat{\boldsymbol{\lambda}}^* - \boldsymbol{\lambda}^*) + O\Big(\|\hat{\boldsymbol{\lambda}}^* - \boldsymbol{\lambda}^*\|_2^2\Big). \tag{8}$$

Substituting Eq. (8) into Eq. (5) gives

$$\Delta \boldsymbol{b} = \nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*) \nabla_{\boldsymbol{\lambda}} \theta(\boldsymbol{\lambda}^*)(\hat{\boldsymbol{\lambda}}^* - \boldsymbol{\lambda}^*) + O\Big(\|\hat{\boldsymbol{\lambda}}^* - \boldsymbol{\lambda}^*\|_2^2\Big). \tag{9}$$

The condition $\nabla_\theta \mathcal{L}(\theta(\boldsymbol{\lambda}), \boldsymbol{\lambda}) = 0$ holds for all $\boldsymbol{\lambda}$ near $\boldsymbol{\lambda}^*$, so by Assumption 3 we may differentiate it with respect to $\boldsymbol{\lambda}$. Applying the chain rule to $\nabla_\theta \mathcal{L}(\theta, \boldsymbol{\lambda}) = \nabla_\theta \mathcal{R}(\theta) - \sum_{i=1}^m \lambda_i \nabla_\theta \mathcal{J}_i(\theta)$ gives $H(\theta(\boldsymbol{\lambda}), \boldsymbol{\lambda}) \nabla_{\boldsymbol{\lambda}} \theta(\boldsymbol{\lambda}) = \nabla_\theta \boldsymbol{\mathcal{J}}(\theta(\boldsymbol{\lambda}))^\top$. Thus, by Assumption 2,

$$\nabla_{\boldsymbol{\lambda}} \theta(\boldsymbol{\lambda}^*) = H(\theta^*, \boldsymbol{\lambda}^*)^{-1} \nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*)^\top. \tag{10}$$

Substituting Eq. (10) into Eq. (9), and establishing that the $m \times m$ matrix $\nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*) H(\theta^*, \boldsymbol{\lambda}^*)^{-1} \nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*)^\top$ is negative definite, and hence invertible, so that $\|\hat{\boldsymbol{\lambda}}^* - \boldsymbol{\lambda}^*\|_2 = O(\|\Delta \boldsymbol{b}\|_2)$, yields the following result.

Theorem 2 (Dual shift approximation)**.** *Under Assumptions 1–5,*

$$\hat{\boldsymbol{\lambda}}^* = \boldsymbol{\lambda}^* + \left(\nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*) H(\theta^*, \boldsymbol{\lambda}^*)^{-1} \nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*)^\top\right)^{-1} \Delta \boldsymbol{b} + O(\|\Delta \boldsymbol{b}\|_2^2). \tag{11}$$

When $m = 1$, this reduces to

$$\hat{\lambda}^* = \lambda^* + \frac{\Delta b}{\nabla_\theta \mathcal{J}(\theta^*)^\top H(\theta^*, \lambda^*)^{-1} \nabla_\theta \mathcal{J}(\theta^*)} + O(\Delta b^2).$$

The complete proof is provided in Appendix B.2. Based on Theorem 2, we define the dual shift approximation $\tilde{\boldsymbol{\lambda}} := \left[\boldsymbol{\lambda}^* + (\nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*) H(\theta^*, \boldsymbol{\lambda}^*)^{-1} \nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*)^\top)^{-1} \Delta \boldsymbol{b}\right]^+$ by dropping the second-order term in Eq. (11) and projecting onto the feasible set $\{\boldsymbol{\lambda} \succeq \mathbf{0}\}$, where $[\cdot]^+$ applies $\max(0, \cdot)$ componentwise.

Theorems 1 and 2 together provide primal-dual shift approximations based on first-order sensitivity analysis. The next section shows how to turn these characterizations into a practical algorithm by estimating the required terms from pre-shift information only.

4 Online Estimation for ZAPD

Applying Theorems 1 and 2 to an observed $\Delta \boldsymbol{b}$ requires the constraint Jacobian $\nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*)$, and, for the dual shift, the $m \times m$ matrix $\nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*) H(\theta^*, \boldsymbol{\lambda}^*)^{-1} \nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*)^\top$. Both are evaluated at the pre-shift solution and must be estimated from pre-shift samples. Throughout this section, k_c denotes the iteration at which the shift is observed, and we use the iterates θ_{k_c} and $\boldsymbol{\lambda}_{k_c}$ in place of θ^* and $\boldsymbol{\lambda}^*$, since the underlying method has converged by then. In this section, we describe practical estimators for each and assemble them into ZAPD.

4.1 Estimating the Primal Shift

The constraint gradients $\nabla_\theta \mathcal{J}_i(\theta_k)$ are the policy gradients at iteration k under each cost objective. Many CRL methods already compute estimates $\widehat{\nabla}_\theta \mathcal{J}_i(\theta_k)$ from the data collected at iteration k, which we use directly. For methods that do not explicitly estimate policy gradients, we partition the trajectories collected at iteration k: one part updates the underlying CRL method, and the other is used to estimate the gradients with the GPOMDP estimator (Baxter & Bartlett, 2001), which requires no value function and can be written as

$$\widehat{\nabla}_\theta \mathcal{J}_i(\theta_k) = \frac{1}{|\mathcal{B}_k|} \sum_{\tau \in \mathcal{B}_k} \sum_{t=0}^{T-1} \gamma^t c_i(s_t^\tau, a_t^\tau) \left(\sum_{h=0}^{t} \nabla_\theta \log \pi_{\theta_k}(a_h^\tau \mid s_h^\tau) \right), \quad i = 1, \ldots, m, \tag{12}$$

where $\mathcal{B}_k$ denotes the batch of trajectories τ used for these estimates, and T is the trajectory horizon. The m estimates differ only in the cost signal c_i weighting each trajectory, so they are computed from the same batch. In either case, ZAPD requires no additional environment interaction, since it reuses data that the underlying method already collects.

To reduce the variance of the estimates, we smooth each with an exponential moving average (EMA):

$$\widetilde{\nabla}_\theta \mathcal{J}_i(\theta_k) = \beta\, \widetilde{\nabla}_\theta \mathcal{J}_i(\theta_{k-1}) + (1 - \beta)\, \widehat{\nabla}_\theta \mathcal{J}_i(\theta_k), \quad \beta \in [0, 1), \quad i = 1, \ldots, m. \tag{13}$$

We write $\widetilde{\nabla}_\theta \boldsymbol{\mathcal{J}}(\theta_k) \in \mathbb{R}^{m\times n}$ for the matrix whose i-th row is $\widetilde{\nabla}_\theta \mathcal{J}_i(\theta_k)^\top$. At the shift iteration k_c, substituting this estimate for $\nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*)$ in Theorem 1 gives a practical primal shift approximation:

$$\tilde{\theta} = \theta_{k_c} + \widetilde{\nabla}_\theta \boldsymbol{\mathcal{J}}(\theta_{k_c})^\top \big(\widetilde{\nabla}_\theta \boldsymbol{\mathcal{J}}(\theta_{k_c}) \widetilde{\nabla}_\theta \boldsymbol{\mathcal{J}}(\theta_{k_c})^\top\big)^{-1} \Delta \boldsymbol{b}. \tag{14}$$

4.2 Estimating the Dual Shift

Theorem 2 additionally requires the $m \times m$ matrix $M = \nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*) H(\theta^*, \boldsymbol{\lambda}^*)^{-1} \nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*)^\top$. The constraint Jacobian is estimated as in Section 4.1, so it remains to estimate how $H(\theta^*, \boldsymbol{\lambda}^*)^{-1}$ acts on each of its rows.

Once $\tilde{\theta}$ is obtained from Eq. (14), a first-order Taylor expansion of $\nabla_\theta \mathcal{L}(\cdot, \boldsymbol{\lambda}^*)$ gives

$$\nabla_\theta \mathcal{L}(\tilde{\theta}, \boldsymbol{\lambda}^*) - \nabla_\theta \mathcal{L}(\theta^*, \boldsymbol{\lambda}^*) \approx H(\theta^*, \boldsymbol{\lambda}^*)(\tilde{\theta} - \theta^*). \tag{15}$$

Defining $y := \nabla_\theta \mathcal{L}(\tilde{\theta}, \boldsymbol{\lambda}^*) - \nabla_\theta \mathcal{L}(\theta^*, \boldsymbol{\lambda}^*)$, Eq. (15) gives $H(\theta^*, \boldsymbol{\lambda}^*)^{-1} y \approx (\tilde{\theta} - \theta^*)$. This single vector equation does not determine the full matrix $H(\theta^*, \boldsymbol{\lambda}^*)^{-1}$, and forming or inverting a Hessian of that size is infeasible for neural policies. We therefore restrict the approximation to a diagonal matrix and fit it by least squares:

$$\min_D \|Dy - (\tilde{\theta} - \theta^*)\|_2^2. \tag{16}$$

Provided $y_l \neq 0$ for all l, the unique solution is $D^* = \operatorname{diag}\Big(\frac{\tilde{\theta}_l - \theta^*_l}{y_l}\Big)$, where l indexes the coordinates of θ. By construction, D^* satisfies the secant equation $D^* y = \tilde{\theta} - \theta^*$, the condition underlying quasi-Newton methods (Nocedal & Wright, 2006).

Computing D^* requires $\tilde{\theta}$, θ^*, and the two gradients $\nabla_\theta \mathcal{L}(\theta^*, \boldsymbol{\lambda}^*)$ and $\nabla_\theta \mathcal{L}(\tilde{\theta}, \boldsymbol{\lambda}^*)$ that define y. Here $\tilde{\theta}$ is given by Eq. (14). The gradient at $\tilde{\theta}$ is not directly available, since ZAPD does not collect $\tau \sim \pi_{\tilde{\theta}}$. We therefore use AlgaeDICE (Nachum et al., 2019), an off-policy policy-gradient estimator, to evaluate both gradients from the pre-shift replay buffer $\mathcal{D}$, so that their difference is formed from a single estimator. We denote its reward and cost gradient estimates at θ by $\nabla_\theta^{\mathrm{AD}} \mathcal{R}(\theta; \mathcal{D})$ and $\nabla_\theta^{\mathrm{AD}} \mathcal{J}_i(\theta; \mathcal{D})$, respectively. Since reliable AlgaeDICE gradient estimates require well-optimized critics, we train its critics on $\mathcal{D}$ during pre-shift training and before estimating the gradients at $\tilde{\theta}$.

The dual shift approximation also uses the GPOMDP constraint Jacobian $\widetilde{\nabla}_\theta \boldsymbol{\mathcal{J}}(\theta_{k_c})$, while $\hat{D}$ is built from AlgaeDICE gradients. Since the two estimators may differ in scale, we calibrate the AlgaeDICE gradients to their pre-shift GPOMDP magnitudes,

$$\kappa_{\mathcal{R}} = \frac{\big\|\widetilde{\nabla}_\theta \mathcal{R}(\theta_{k_c})\big\|_2}{\big\|\nabla_\theta^{\mathrm{AD}} \mathcal{R}(\theta_{k_c}; \mathcal{D})\big\|_2}, \qquad \kappa_{\mathcal{J}_i} = \frac{\big\|\widetilde{\nabla}_\theta \mathcal{J}_i(\theta_{k_c})\big\|_2}{\big\|\nabla_\theta^{\mathrm{AD}} \mathcal{J}_i(\theta_{k_c}; \mathcal{D})\big\|_2}, \quad i = 1, \ldots, m,$$

where the smoothed reward gradient $\widetilde{\nabla}_\theta \mathcal{R}(\theta_{k_c})$ is obtained from Eqs. (12) and (13) by replacing $c_i(\cdot, \cdot)$ with $r(\cdot, \cdot)$. The calibrated gradient difference is

$$\begin{aligned}\hat{y} = {} & \kappa_{\mathcal{R}} \big(\nabla_\theta^{\mathrm{AD}} \mathcal{R}(\tilde{\theta}; \mathcal{D}) - \nabla_\theta^{\mathrm{AD}} \mathcal{R}(\theta_{k_c}; \mathcal{D})\big) \\ & - \sum_{j=1}^{m} \lambda_{k_c, j}\, \kappa_{\mathcal{J}_j} \big(\nabla_\theta^{\mathrm{AD}} \mathcal{J}_j(\tilde{\theta}; \mathcal{D}) - \nabla_\theta^{\mathrm{AD}} \mathcal{J}_j(\theta_{k_c}; \mathcal{D})\big).\end{aligned} \tag{17}$$

Replacing y with $\hat{y}$ in the diagonal approximation gives $\hat{D} = \operatorname{diag}\Big(\frac{\tilde{\theta}_l - \theta_{k_c, l}}{\hat{y}_l}\Big)$. To keep this well defined when a coordinate of $\hat{y}$ is close to zero, we floor its magnitude while preserving its sign, replacing $\hat{y}_l$ by $\operatorname{sign}(\hat{y}_l) \max\{|\hat{y}_l|, \epsilon\}$ with $\epsilon = 10^{-4}$. Substituting $\hat{D}$ for $H(\theta^*, \boldsymbol{\lambda}^*)^{-1}$ in Theorem 2 yields the practical dual shift approximation:

$$\tilde{\boldsymbol{\lambda}} \;=\; \Big[\boldsymbol{\lambda}_{k_c} + \big(\widetilde{\nabla}_\theta \boldsymbol{\mathcal{J}}(\theta_{k_c})\, \hat{D}\, \widetilde{\nabla}_\theta \boldsymbol{\mathcal{J}}(\theta_{k_c})^\top\big)^{-1} \Delta \boldsymbol{b}\Big]^+, \tag{18}$$

where $[\cdot]^+$ applies $\max(0, \cdot)$ componentwise.

The two updates are independent and may be applied separately. The primal update reuses gradients the underlying method already computes, whereas the dual update additionally requires AlgaeDICE

critics and gradient evaluations at $\tilde{\theta}$. Applying the dual update is therefore worthwhile whenever the extra computation is available, and we recommend applying both; Section 5 compares the primal-only and primal-dual variants directly. It can be omitted in two cases. First, it is unnecessary for methods whose multipliers do not depend on their previous values: CVPO (Liu et al., 2022), for instance, solves a convex dual problem determined by θ_k alone at each iteration, so stale multipliers are overwritten rather than carried forward. Second, when computation is limited, applying only the primal update avoids the AlgaeDICE critic training and the additional gradient evaluations at $\tilde{\theta}$.

4.3 Pseudocode

Algorithm 1 summarizes ZAPD. During pre-shift training, we store trajectories in the replay buffer, update the AlgaeDICE critics, and, after K_{EMA}, update the EMA gradients with Eq. (13). Upon a threshold shift, we remove constraints that fail the practical checks corresponding to Assumptions 4 and 5. We apply the primal and dual corrections Eqs. (14) and (18) to the remaining constraints.

Algorithm 1 ZAPD

Require: Iterations K; initial parameters $\theta_0, \boldsymbol{\lambda}_0$; initial thresholds $\boldsymbol{b}_0 = \boldsymbol{b}$; EMA start iteration K_{EMA}; replay buffer $\mathcal{D}$

for $k = 1$ **to** K **do**
  Observe current thresholds $\boldsymbol{b}_k$
  if $\boldsymbol{b}_k = \boldsymbol{b}_{k-1}$ **then**
    $(\theta_k, \boldsymbol{\lambda}_k) \leftarrow (\theta_{k-1}, \boldsymbol{\lambda}_{k-1})$ with standard primal-dual methods
    Store trajectories in $\mathcal{D}$; update AlgaeDICE critics
    if $k \geq K_{\mathrm{EMA}}$ **then**
      Update EMA reward and cost gradients with Eq. (13)
    end if
  else
    Let $\mathcal{I}_k$ be a maximal subset of constraints that passes Assumptions 4 and 5 checks
    if $\mathcal{I}_k \neq \emptyset$ **then**
      $\theta_k \leftarrow \tilde{\theta}$ given by Eq. (14) using constraints in $\mathcal{I}_k$
      On $\mathcal{D}$, estimate AlgaeDICE gradients at θ_{k-1}; update critics; estimate AlgaeDICE gradients at $\tilde{\theta}$
      $\lambda_{k,i} \leftarrow \tilde{\lambda}_i$ given by Eq. (18) for $i \in \mathcal{I}_k$; $\lambda_{k,j} \leftarrow \lambda_{k-1,j}$ for $j \notin \mathcal{I}_k$
    else
      $(\theta_k, \boldsymbol{\lambda}_k) \leftarrow (\theta_{k-1}, \boldsymbol{\lambda}_{k-1})$ with standard primal-dual methods
    end if
  end if
end for

5 Experiments

We organize our experiments around two questions: **Q1: Can ZAPD adapt better and converge faster than threshold-adaptive methods?** and **Q2: Does it remain effective across primal-dual methods and policy parameterizations?**

5.1 Experimental Setup and Metrics

Environments Throughout, we adopt the environments used by the paper introducing the method we compare against or build on, so that each method is evaluated on the domain it was designed for. The specific environments are introduced in Sections 5.2 and 5.3, with full details in Appendix C.1. Each environment defines a single cost, so all main text experiments have $m = 1$. All main text experiments use 20 independent runs.

Threshold Settings Before the shift, each method has access to a set $\mathcal{E}$ of available thresholds, and we refer to $|\mathcal{E}| = 1, 2, 3$ as the single-, double-, and triple-threshold settings. The single-threshold setting is the one ZAPD is designed for, as it requires only one converged solution, while the double- and triple-threshold settings are those the threshold-adaptive baselines are designed for.

When ZAPD is evaluated in the multiple-threshold settings, it independently trains one converged solution per available threshold, each with the single-threshold training budget, and applies its zero-shot update from the one closest to the post-shift threshold. Within each setting, all methods are given the same available thresholds and the same total pre-shift interaction budget. Full details are provided in Appendix C.2.

Metrics The *immediate approximation error* measures the post-shift accuracy after one update. We freeze the resulting policy, average its reward and cost over 50 episodes, and compute their relative errors with respect to the post-shift reference values under $\hat{b}$. We report the average of the two errors. The *first-convergence iteration* is the first iteration at which both reward and cost are within 5% of their post-shift reference values. For each threshold shift, P_{50} and P_{80} denote the iterations by which 50% and 80% of the 20 runs have first converged, respectively. We report P_{50} and P_{80} averaged over threshold shifts. Lower is better for both metrics. Metric details are provided in Appendix C.3. Due to space limitations, we only present aggregated results in the main text. Please see Appendix D for the detailed breakdowns.

5.2 Q1: Can ZAPD Adapt Better and Converge Faster?

We compare ZAPD with CCPO (Yao et al., 2023) and CAPS (Chemingui et al., 2025) on four tasks from the Bullet Safety Gym benchmark (Gronauer, 2022): Ball-Circle, Car-Circle, Drone-Circle, and Drone-Run. We use the same tasks and constraint thresholds as Yao et al. (2023), and use CVPO (Liu et al., 2022), the basis of CCPO, as the underlying CRL method for ZAPD. Since CVPO recomputes its multipliers from the current policy at every iteration rather than carrying them forward, the dual update is unnecessary here, and we apply the primal-only ZAPD (Section 4.2).

Immediate Approximation Error Table 1 reports the immediate approximation error. In the single-threshold setting ZAPD achieves the lowest error on all four tasks, reducing error by 30% on average relative to the better baseline. The gap is structural rather than incidental: with a single available threshold, CCPO has no range to condition across and CAPS has a one-entry library, so neither can exploit the mechanism it was designed around. The double- and triple-threshold settings give the baselines the multi-threshold access they are built around, and there ZAPD still attains the lowest average error across tasks, by 23% and 9% relative to the better baseline.

Table 1: Immediate approximation average error (%) for the Bullet Safety Gym tasks at $\hat{b} \in \{5, 10, \ldots, 75\}$. Each cell reports mean $\pm$ standard deviation. Dark shading indicates the lowest mean.

	Ball-Circle			Car-Circle		
Setting	ZAPD	CCPO	CAPS	ZAPD	CCPO	CAPS
Single	**19.33** $\pm$ 6.27	58.82 $\pm$ 2.40	53.55 $\pm$ 2.57	**47.35** $\pm$ 5.56	67.87 $\pm$ 14.03	51.48 $\pm$ 6.35
Double	**14.67** $\pm$ 4.78	59.75 $\pm$ 40.35	37.10 $\pm$ 1.97	36.77 $\pm$ 2.41	**32.71** $\pm$ 9.45	41.66 $\pm$ 7.95
Triple	**13.36** $\pm$ 2.82	34.60 $\pm$ 32.29	23.61 $\pm$ 3.46	30.19 $\pm$ 4.01	**21.36** $\pm$ 3.82	25.89 $\pm$ 8.50
	Drone-Circle			Drone-Run		
Setting	ZAPD	CCPO	CAPS	ZAPD	CCPO	CAPS
Single	**25.23** $\pm$ 5.21	65.89 $\pm$ 3.94	45.00 $\pm$ 2.93	**46.47** $\pm$ 23.76	95.89 $\pm$ 27.07	48.55 $\pm$ 33.65
Double	**21.80** $\pm$ 3.10	31.90 $\pm$ 2.76	31.84 $\pm$ 4.60	43.88 $\pm$ 21.24	79.77 $\pm$ 37.82	**41.48** $\pm$ 18.80
Triple	**18.48** $\pm$ 2.24	21.25 $\pm$ 2.88	21.30 $\pm$ 4.50	**36.57** $\pm$ 12.63	83.73 $\pm$ 41.10	37.14 $\pm$ 16.75

First-Convergence Iteration We report first-convergence results for the single-threshold setting. Table 2 reports P_{50} and P_{80} averaged over post-shift thresholds. ZAPD reaches P_{50} fastest on all four tasks and P_{80} fastest on three, with the largest margin on Drone-Circle, where it converges more than twice as fast as the better baseline. The improved one-step approximation of Table 1 therefore translates into faster reconvergence, not merely a better starting point.

Table 2: First-convergence iterations (P_{50}/P_{80}, averaged over $\hat{b}$) for the Bullet Safety Gym tasks. Bold indicates the lowest value for each percentile and dark shading the lowest for both.

Environment	ZAPD	CCPO	CAPS
Ball-Circle	**7.88 / 9.15**	20.69 / 29.35	11.12 / 12.98
Car-Circle	**11.50 / 15.68**	23.62 / 29.98	14.44 / 21.10
Drone-Circle	**13.19 / 19.18**	53.12 / 60.25	35.88 / 42.75
Drone-Run	**29.0** / 63.1	45.69 / 66.10	30.81 / **52.82**

5.3 Q2: Is ZAPD Effective Across Methods and Policies?

We evaluate ZAPD as a plug-in for two primal-dual CRL methods, C-PG (Montenegro et al., 2024) and PPO-Lagrangian (Ji et al., 2024) (PPO-L), using linear and neural network (NN) policies. Following the same principle as above, we evaluate on the domains these methods are studied in: CostLQR, using the setting and cost definition of Montenegro et al. (2024), and the Swimmer task `SafetySwimmerVelocity-v1` from Safety-Gymnasium (Ji et al., 2023). For these two environments we select the threshold ranges ourselves. We compare ZAPD only against CAPS, since this section evaluates plug-in use with given base methods, whereas CCPO replaces the base method rather than building on it.

Immediate Approximation Error Table 3 reports the immediate approximation error. ZAPD achieves lower or equal error than CAPS in all but one of the 24 comparisons, reducing error by 31% on average. The margin is consistent across both base methods and both policy parameterizations, indicating that the sensitivity correction does not depend on how the underlying method represents its policy or computes its updates.

Table 3: Immediate approximation average error (%) for CostLQR and Swimmer, with $\hat{b}$ ranging over $[6, 9]$ and $[1, 4]$, respectively. Each cell reports mean $\pm$ standard deviation. Dark shading indicates the lowest mean.

		C-PG (Linear)		PPO-L (Linear)		C-PG (NN)		PPO-L (NN)	
Environment	Setting	ZAPD	CAPS	ZAPD	CAPS	ZAPD	CAPS	ZAPD	CAPS
CostLQR	Single	**5.2** $\pm$ 2.7	8.5 $\pm$ 0.9	**7.6** $\pm$ 2.9	10.3 $\pm$ 2.9	**9.6** $\pm$ 5.1	11.0 $\pm$ 2.4	**9.2** $\pm$ 5.0	10.3 $\pm$ 4.0
	Double	**4.3** $\pm$ 1.9	7.7 $\pm$ 2.3	**5.3** $\pm$ 2.0	8.0 $\pm$ 1.8	**7.6** $\pm$ 3.9	10.4 $\pm$ 2.9	**6.2** $\pm$ 2.4	8.3 $\pm$ 1.6
	Triple	**4.4** $\pm$ 1.9	5.6 $\pm$ 2.4	**5.7** $\pm$ 2.0	6.6 $\pm$ 2.2	**7.8** $\pm$ 4.0	8.6 $\pm$ 3.2	**5.9** $\pm$ 2.8	6.8 $\pm$ 2.1
Swimmer	Single	**18.1** $\pm$ 9.5	29.1 $\pm$ 9.2	**15.8** $\pm$ 8.5	30.1 $\pm$ 2.3	**26.1** $\pm$ 17.0	34.4 $\pm$ 14.8	**10.2** $\pm$ 3.8	30.5 $\pm$ 2.3
	Double	**14.9** $\pm$ 6.2	17.2 $\pm$ 3.8	**5.0** $\pm$ 1.5	19.4 $\pm$ 0.7	**22.9** $\pm$ 16.2	27.5 $\pm$ 6.6	**5.0** $\pm$ 1.6	14.5 $\pm$ 0.7
	Triple	15.3 $\pm$ 6.5	**14.9** $\pm$ 4.5	**4.9** $\pm$ 1.6	13.2 $\pm$ 1.2	**24.8** $\pm$ 18.2	**24.8** $\pm$ 8.4	**4.4** $\pm$ 1.7	11.2 $\pm$ 0.9

First-Convergence Iteration Table 4 reports P_{50} and P_{80} averaged over post-shift thresholds in the single-threshold setting, comparing ZAPD (Primal), which applies only the primal update, ZAPD (Full), which applies both, and CAPS. Both variants converge faster than CAPS in 15 of the 16 percentile comparisons, the exception being Swimmer with an NN policy under C-PG. ZAPD (Full) improves on ZAPD (Primal) in 13 of the 16, often substantially: on Swimmer with an NN policy under PPO-L it reaches P_{50} nearly three times as fast. This confirms that the dual correction of Theorem 2 contributes beyond the primal update alone.

We additionally evaluate ZAPD under simultaneous shifts of multiple constraints in Appendix E.

Table 4: First-convergence iterations (P_{50}/P_{80}, averaged over $\hat{b}$) for CostLQR and Swimmer. Bold indicates the lowest value for each percentile and dark shading the lowest for both.

		C-PG			PPO-L		
Policy	Environment	ZAPD (Full)	ZAPD (Primal)	CAPS	ZAPD (Full)	ZAPD (Primal)	CAPS
Linear	CostLQR	4.1 / **36.1**	**3.5** / 37.9	78.5 / 156.1	**4.8 / 13.8**	5.4 / 30.8	18.1 / 33.0
	Swimmer	42.8 / **171.2**	**36.5** / 244.0	201.8 / 385.8	**30.0 / 52.3**	49.0 / 83.0	95.3 / 112.8
NN	CostLQR	**12.2 / 34.1**	13.9 / 55.9	40.2 / 93.9	**4.8 / 14.1**	7.1 / 18.1	15.5 / 25.2
	Swimmer	**224.0** / 870.7	379.2 / 787.7	392.7 / **690.8**	**16.7 / 53.3**	48.2 / 92.0	89.8 / 107.7

6 Conclusions

We introduce ZAPD for abrupt constraint-threshold shifts in CRL. Using first-order sensitivity analysis, ZAPD derives closed-form primal and dual zero-shot updates from pre-shift information and the observed threshold shift. Experiments show that ZAPD provides improved post-shift initialization and faster reconvergence than threshold-adaptive baselines. As a plug-in to primal-dual CRL methods, ZAPD remains effective across different methods and policy parameterizations. We also find that the primal-only update is often effective, while applying both typically provides further improvement. As a first-order method, ZAPD may become less accurate for large threshold shifts.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools to formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology and experiments, implement methods, develop theoretical models or conceptual frameworks, and interpret results. We did not use generative AI tools to generate synthetic datasets, or assist with translation. Dataset cleaning and reformatting and qualitative or thematic data analysis are not applicable to this work.

Additionally, we used generative AI tools to create or edit software code, draft parts of the appendix, edit and format parts of the paper including tables and figures in LaTeX, identify relevant literature, and contribute to brainstorming the acronym "ZAPD."

We reviewed all AI-assisted work. AI-assisted mathematical claims, proof ingredients, proofs, hypotheses, and theoretical or conceptual development were manually checked by two authors. AI-assisted experimental design, method implementation, and interpretation of results were manually checked by one author by examining the raw experimental results and maintaining explicit experimental-design specifications in Markdown files. AI-assisted Python code was manually checked by one author against these experimental-design specifications and the resulting raw data. AI-assisted appendix text was manually checked by all three authors. AI-assisted LaTeX tables and figures were compiled and manually inspected to verify that they matched the intended presentation and complied with the ICLR formatting requirements. Literature identified with AI assistance was manually verified by two authors to confirm the existence and relevance of the cited papers. Generative AI indirectly contributed to brainstorming the acronym "ZAPD," and we manually checked for prior use of the acronym.

We take responsibility for the final content of this work, including text, claims, code, and other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

(This section is **recommended** and does not count toward the page limit.)

We are not aware of any specific ethical concerns arising from this work.

REPRODUCIBILITY STATEMENT

(This section is **recommended** and does not count toward the page limit.)

We provide the assumptions and theoretical derivations underlying ZAPD in Section 3, with complete proofs of Theorems 1 and 2 in Appendix B. Section 4 describes the practical estimation procedure, and Algorithm 1 provides the pseudocode. The experimental setup and evaluation protocol are described in Section 5.1, with further details on the environments, evaluation metrics, threshold settings and interaction budgets, and implementation details in Appendix C. Anonymous source code for reproducing the experiments is provided in this link.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International Conference on Machine Learning (ICML)*, pp. 22–31, 2017.

Eitan Altman. *Constrained Markov decision processes*. Routledge, 2021.

Wensong Bai, Chao Zhang, Qihang Xu, Chufan Chen, Chenhao Zhou, and Hui Qian. Conditional sequence modeling for safe reinforcement learning. *Mathematics*, 14(6):1015, 2026.

Jonathan Baxter and Peter L Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.

Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, second edition, 1999.

V.S. Borkar. An actor-critic algorithm for constrained Markov decision processes. *Systems & Control Letters*, 54(3):207–213, 2005.

Nicolas Carrara, Edouard Leurent, Romain Laroche, Tanguy Urvoy, Odalric-Ambrym Maillard, and Olivier Pietquin. Budgeted reinforcement learning in continuous state space. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 9299–9309, 2019.

Yassine Chemingui, Aryan Deshwal, Honghao Wei, Alan Fern, and Jana Doppa. Constraint-adaptive policy switching for offline safe reinforcement learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, pp. 15722–15730, 2025.

Dongsheng Ding, Kaiqing Zhang, Tamer Basar, and Mihailo Jovanovic. Natural policy gradient primal-dual method for constrained Markov decision processes. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 8378–8390, 2020.

Dongsheng Ding, Chen-Yu Wei, Kaiqing Zhang, and Alejandro Ribeiro. Last-iterate convergent policy gradient primal-dual methods for constrained MDPs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 66138–66200, 2023.

Yuhao Ding and Javad Lavaei. Provably efficient primal-dual reinforcement learning for CMDPs with non-stationary objectives and constraints. In *AAAI Conference on Artificial Intelligence (AAAI)*, pp. 7396–7404, 2023.

Anthony V Fiacco. Sensitivity analysis for nonlinear programming using penalty methods. *Mathematical Programming*, 10(1):287–311, 1976.

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

Sven Gronauer. Bullet-Safety-Gym: A framework for constrained reinforcement learning. Technical report, mediaTUM, 2022.

Zijian Guo, Weichao Zhou, Shengao Wang, and Wenchao Li. Constraint-conditioned actor-critic for offline safe reinforcement learning. In *International Conference on Learning Representations (ICLR)*, pp. 41581–41602, 2025.

I-Hong Hou. Network optimization in dynamic systems: Fast adaptation via zero-shot Lagrangian update. In *IEEE Conference on Computer Communications (INFOCOM)*, pp. 1–10, 2025.

Weidong Huang, Jiaming Ji, Borong Zhang, Chunhe Xia, and Yaodong Yang. SafeDreamer: Safe reinforcement learning with world models. In *International Conference on Learning Representations (ICLR)*, 2024.

Jiaming Ji, Borong Zhang, Jiayi Zhou, Xuehai Pan, Weidong Huang, Ruiyang Sun, Yiran Geng, Yifan Zhong, Josef Dai, and Yaodong Yang. Safety Gymnasium: A unified safe reinforcement learning benchmark. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023.

Jiaming Ji, Jiayi Zhou, Borong Zhang, Juntao Dai, Xuehai Pan, Ruiyang Sun, Weidong Huang, Yiran Geng, Mickel Liu, and Yaodong Yang. OmniSafe: An infrastructure for accelerating safe reinforcement learning research. *Journal of Machine Learning Research*, 25(285):1–6, 2024.

Qian Lin, Bo Tang, Zifan Wu, Chao Yu, Shangqin Mao, Qianlong Xie, Xingxing Wang, and Dong Wang. Safe offline reinforcement learning with real-time budget constraints. In *International Conference on Machine Learning (ICML)*, 2023.

Yongshuai Liu, Jiaxin Ding, and Xin Liu. IPO: Interior-point policy optimization under constraints. In *AAAI Conference on Artificial Intelligence (AAAI)*, pp. 4940–4947, 2020.

Zifan Liu, Xinran Li, and Jun Zhang. C2IQL: Constraint-conditioned implicit Q-learning for safe offline reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2025.

Zuxin Liu, Zhepeng Cen, Vladislav Isenbaev, Wei Liu, Steven Wu, Bo Li, and Ding Zhao. Constrained variational policy optimization for safe reinforcement learning. In *International Conference on Machine Learning (ICML)*, pp. 13644–13668, 2022.

Zuxin Liu, Zijian Guo, Yihang Yao, Zhepeng Cen, Wenhao Yu, Tingnan Zhang, and Ding Zhao. Constrained decision transformer for offline safe reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2023.

Yuxiao Lu, Arunesh Sinha, and Pradeep Varakantham. Handling long and richly constrained tasks through constrained hierarchical reinforcement learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2024.

Alessandro Montenegro, Marco Mussi, Matteo Papini, and Alberto Maria Metelli. Last-iterate global convergence of policy gradients for constrained reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 126363–126416, 2024.

Ofir Nachum, Bo Dai, Ilya Kostrikov, Yinlam Chow, Lihong Li, and Dale Schuurmans. AlgaeDICE: Policy gradient from arbitrary experience. *arXiv preprint arXiv:1912.02074*, 2019.

Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 2006.

Santiago Paternain, Luiz Chamon, Miguel Calvo-Fullana, and Alejandro Ribeiro. Constrained reinforcement learning has zero duality gap. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning, 2019. URL `https://cdn.openai.com/safexp-short.pdf`.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Peter Stechlinski, Kamil A. Khan, and Paul I. Barton. Generalized sensitivity analysis of nonlinear programs. *SIAM Journal on Optimization*, 28(1):272–301, 2018.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by PID Lagrangian methods. In *International Conference on Machine Learning (ICML)*, pp. 9133–9143, 2020.

Francesco Emanuele Stradi, Anna Lunghi, Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. Taming adversarial constraints in CMDPs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 9697–9745, 2025.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT Press, 1998.

Chen Tessler, Daniel J Mankowitz, and Shie Mannor. Reward constrained policy optimization. In *International Conference on Learning Representations (ICLR)*, 2019.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5026–5033, 2012.

Honghao Wei, Arnob Ghosh, Ness Shroff, Lei Ying, and Xingyu Zhou. Provably efficient model-free algorithms for non-stationary CMDPs. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 6527–6570, 2023.

Tsung-Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J. Ramadge. Projection-based constrained policy optimization. In *International Conference on Learning Representations (ICLR)*, 2020.

Yihang Yao, Zuxin Liu, Zhepeng Cen, Jiacheng Zhu, Wenhao Yu, Tingnan Zhang, and Ding Zhao. Constraint-conditioned policy optimization for versatile safe reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 12555–12568, 2023.

Jiahui Zhu, Kihyun Yu, Dabeen Lee, Xin Liu, and Honghao Wei. An optimistic algorithm for online CMDPs with anytime adversarial constraints. In *International Conference on Machine Learning (ICML)*, pp. 80347–80372, 2025.

A Related Work

Adaptive constrained reinforcement learning. Most constrained reinforcement learning methods optimize a policy for a constraint threshold fixed during training. A smaller body of work instead seeks policies that can operate under different safety budgets. Budgeted reinforcement learning augments the decision problem with the current budget and learns a policy over a range of admissible budgets (Carrara et al., 2019). Related approaches in offline safe RL learn constraint-conditioned policies or trajectory generators, enabling adaptation to different constraint thresholds at deployment (Lin et al., 2023; Liu et al., 2023; Guo et al., 2025; Liu et al., 2025; Bai et al., 2026). CoSHRL combines hierarchical RL with constrained search to accommodate flexible constraint thresholds without retraining (Lu et al., 2024). These approaches complement threshold-conditioned methods such as CCPO and policy-switching methods such as CAPS (Yao et al., 2023; Chemingui et al., 2025). However, these methods generally obtain threshold adaptivity by incorporating multiple budgets during training, conditioning their learned components on budget information, or retaining structures that support adaptation across a range of thresholds. ZAPD considers a different setting: the agent has converged under one threshold, an unseen threshold shift is then observed, and adaptation is performed directly from the pre-shift solution without additional environment interaction.

Non-stationary constrained reinforcement learning. A related line of research studies non-stationary CMDPs in which reward functions, cost functions, or transition dynamics evolve over time. Representative algorithms use periodic restarting and optimism to adapt to such non-stationarity while providing guarantees on dynamic regret and constraint violation (Ding & Lavaei, 2023; Wei et al., 2023). More recent work considers non-stationary reward and constraint sequences and derives performance guarantees that degrade smoothly with the degree of environmental non-stationarity (Stradi et al., 2025). These works address broader non-stationarity through continued interaction. ZAPD instead isolates an abrupt threshold shift with fixed reward, cost, and dynamics, enabling an immediate update from the pre-shift solution before collecting post-shift samples.

Sensitivity analysis and fast adaptation. Our approach is also related to sensitivity analysis in parametric optimization, which studies how primal and dual solutions vary under parameter perturbations (Fiacco, 1976) and has been extended to settings with changes in the active set (Stechlinski et al., 2018). Such sensitivity arguments have also been used for rapid adaptation in dynamic network optimization, where zero-shot updates of Lagrange multipliers can be derived from information available at the time of an abrupt system change (Hou, 2025). ZAPD brings this perspective to constrained reinforcement learning.

B Proofs

B.1 Theorem 1

Proof. Write $J := \nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*)^\top \in \mathbb{R}^{n\times m}$ for the matrix whose i-th column is the constraint gradient $\nabla_\theta \mathcal{J}_i(\theta^*)$, so that the constrained optimization problem in Eq. (6) reads

$$\min_\theta \ \frac{1}{2}\|\theta-\theta^*\|_2^2 \quad \text{subject to} \quad J^\top(\theta-\theta^*) = \Delta\boldsymbol{b}. \tag{19}$$

With a vector of Lagrange multipliers $\boldsymbol{\nu} \in \mathbb{R}^m$, the Lagrangian of Eq. (19) is

$$F(\theta,\boldsymbol{\nu}) = \frac{1}{2}\|\theta-\theta^*\|_2^2 - \boldsymbol{\nu}^\top\big(J^\top(\theta-\theta^*) - \Delta\boldsymbol{b}\big). \tag{20}$$

Setting the gradient with respect to θ to zero gives $\nabla_\theta F = (\theta-\theta^*) - J\boldsymbol{\nu} = 0$, and therefore

$$\theta - \theta^* = J\boldsymbol{\nu}. \tag{21}$$

Setting the gradient with respect to $\boldsymbol{\nu}$ to zero recovers the constraint $J^\top(\theta-\theta^*) = \Delta\boldsymbol{b}$. Substituting Eq. (21) into it yields the $m \times m$ linear system

$$J^\top J\,\boldsymbol{\nu} = \Delta\boldsymbol{b}. \tag{22}$$

By Assumption 5, $\nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*)$ has full row rank, so J has full column rank and the Gram matrix $J^\top J$ is positive definite, hence invertible. Consequently Eq. (22) has the unique solution

$$\boldsymbol{\nu} = (J^\top J)^{-1}\Delta\boldsymbol{b}. \tag{23}$$

Full column rank of J also implies that $J^\top$ is surjective, so the affine constraint in Eq. (19) is feasible for every $\Delta \boldsymbol{b}$. Since the objective is strictly convex and the feasible set is affine, the minimizer is unique. Substituting Eq. (23) into Eq. (21), this unique solution is

$$\tilde{\theta} = \theta^* + J\,(J^\top J)^{-1}\,\Delta \boldsymbol{b}, \tag{24}$$

which is Eq. (7) written in terms of J, since $J(J^\top J)^{-1}$ is the Moore–Penrose pseudoinverse of $J^\top$. For $m = 1$ the matrix J reduces to the single column $\nabla_\theta \mathcal{J}(\theta^*)$ and $J^\top J$ to the scalar $\|\nabla_\theta \mathcal{J}(\theta^*)\|_2^2$, giving the special case stated in Theorem 1. □

B.2 Theorem 2

Proof. As in the proof of Theorem 1, write $J := \nabla_\theta \boldsymbol{\mathcal{J}}(\theta^*)^\top \in \mathbb{R}^{n\times m}$ for the matrix whose columns are the constraint gradients, and abbreviate $H := H(\theta^*, \boldsymbol{\lambda}^*)$.

By the definition of $\theta(\boldsymbol{\lambda})$ in Assumption 3, $\nabla_\theta \mathcal{L}(\theta(\boldsymbol{\lambda}), \boldsymbol{\lambda}) = 0$ for all $\boldsymbol{\lambda}$ near $\boldsymbol{\lambda}^*$, and by Assumption 2 this stationary point is locally unique, so $\hat{\theta}^* = \theta(\hat{\boldsymbol{\lambda}}^*)$ and $\theta^* = \theta(\boldsymbol{\lambda}^*)$. A first-order Taylor expansion of $\theta(\boldsymbol{\lambda})$ around $\boldsymbol{\lambda}^*$ therefore gives

$$\hat{\theta}^* - \theta^* = \nabla_{\boldsymbol{\lambda}} \theta(\boldsymbol{\lambda}^*)(\hat{\boldsymbol{\lambda}}^* - \boldsymbol{\lambda}^*) + O\Big(\|\hat{\boldsymbol{\lambda}}^* - \boldsymbol{\lambda}^*\|_2^2\Big). \tag{25}$$

By Assumption 4, $\boldsymbol{\mathcal{J}}(\theta^*) = \boldsymbol{b}$ and $\boldsymbol{\mathcal{J}}(\hat{\theta}^*) = \hat{\boldsymbol{b}}$. Subtracting these and expanding $\boldsymbol{\mathcal{J}}(\hat{\theta}^*)$ around θ^* gives

$$\Delta \boldsymbol{b} = J^\top(\hat{\theta}^* - \theta^*) + O\Big(\|\hat{\theta}^* - \theta^*\|_2^2\Big). \tag{26}$$

Eq. (25) implies $\|\hat{\theta}^* - \theta^*\|_2 = O(\|\hat{\boldsymbol{\lambda}}^* - \boldsymbol{\lambda}^*\|_2)$, so substituting Eq. (25) into Eq. (26) yields

$$\Delta \boldsymbol{b} = J^\top \nabla_{\boldsymbol{\lambda}} \theta(\boldsymbol{\lambda}^*)(\hat{\boldsymbol{\lambda}}^* - \boldsymbol{\lambda}^*) + O\Big(\|\hat{\boldsymbol{\lambda}}^* - \boldsymbol{\lambda}^*\|_2^2\Big). \tag{27}$$

To evaluate the Jacobian, note that $\nabla_\theta \mathcal{L}(\theta(\boldsymbol{\lambda}), \boldsymbol{\lambda}) = 0$ holds for all $\boldsymbol{\lambda}$ near $\boldsymbol{\lambda}^*$, so we may differentiate it with respect to $\boldsymbol{\lambda}$. Applying the chain rule to $\nabla_\theta \mathcal{L}(\theta, \boldsymbol{\lambda}) = \nabla_\theta \mathcal{R}(\theta) - \sum_{i=1}^m \lambda_i \nabla_\theta \mathcal{J}_i(\theta)$ gives

$$H(\theta(\boldsymbol{\lambda}), \boldsymbol{\lambda}) \nabla_{\boldsymbol{\lambda}} \theta(\boldsymbol{\lambda}) = \nabla_\theta \boldsymbol{\mathcal{J}}(\theta(\boldsymbol{\lambda}))^\top, \tag{28}$$

where the j-th column of each side is the derivative with respect to λ_j. By Assumption 2, H is invertible, so

$$\nabla_{\boldsymbol{\lambda}} \theta(\boldsymbol{\lambda}^*) = H^{-1} J. \tag{29}$$

Substituting Eq. (29) into Eq. (27) gives

$$\Delta \boldsymbol{b} = M(\hat{\boldsymbol{\lambda}}^* - \boldsymbol{\lambda}^*) + O\Big(\|\hat{\boldsymbol{\lambda}}^* - \boldsymbol{\lambda}^*\|_2^2\Big), \qquad M := J^\top H^{-1} J \in \mathbb{R}^{m\times m}. \tag{30}$$

Since θ^* maximizes $\mathcal{L}(\cdot, \boldsymbol{\lambda}^*)$, the Hessian H is negative semidefinite, and Assumption 2 makes it invertible, hence negative definite. By Assumption 5, J has full column rank, so for any $\boldsymbol{v} \neq \mathbf{0}$ we have $J\boldsymbol{v} \neq \mathbf{0}$ and therefore $\boldsymbol{v}^\top M \boldsymbol{v} = (J\boldsymbol{v})^\top H^{-1}(J\boldsymbol{v}) < 0$. Thus M is negative definite and in particular invertible. Consequently Eq. (30) implies $\|\hat{\boldsymbol{\lambda}}^* - \boldsymbol{\lambda}^*\|_2 = O(\|\Delta \boldsymbol{b}\|_2)$, so the remainder in Eq. (30) is $O(\|\Delta \boldsymbol{b}\|_2^2)$. Solving for the dual shift gives

$$\hat{\boldsymbol{\lambda}}^* = \boldsymbol{\lambda}^* + M^{-1}\Delta \boldsymbol{b} + O\big(\|\Delta \boldsymbol{b}\|_2^2\big), \tag{31}$$

which is Eq. (11) written in terms of J and H. For $m = 1$ the matrix M reduces to the scalar $\nabla_\theta \mathcal{J}(\theta^*)^\top H^{-1} \nabla_\theta \mathcal{J}(\theta^*)$, giving the special case stated in Theorem 2. □

C Experimental Details

C.1 Environments

Throughout this section, d_S and d_A denote the state and action dimensions, respectively, and T denotes the episode horizon.

Bullet Safety Gym. We use four tasks from the Bullet Safety Gym benchmark (Gronauer, 2022): Ball-Circle, Car-Circle, Drone-Circle, and Drone-Run. Their state and action dimensions and episode horizons are

Environment	d_S	d_A	T
Ball-Circle	8	2	200
Car-Circle	8	2	300
Drone-Circle	18	4	300
Drone-Run	17	4	200

(32)

For the Circle tasks, let $\mathbf{x}_t = [x_t, y_t]^\top$ and $\mathbf{v}_t = [v_{x,t}, v_{y,t}]^\top$ denote the agent position and velocity. The reward and cost are defined as

$$r_t = \frac{-y_t v_{x,t} + x_t v_{y,t}}{1 + |\|\mathbf{x}_t\|_2 - r_{\text{circle}}|} + r_{\text{robot},t}, \qquad c_t = \mathbf{1}(|x_t| > x_{\text{lim}}), \tag{33}$$

where r_{circle} is the target-circle radius, x_{lim} specifies the safe region, and $r_{\text{robot},t}$ is the robot-specific reward term.

For the Run task, the agent moves along a straight corridor. Let $\mathbf{g}$ denote the position of a fictitious target. The reward and cost are defined as

$$r_t = \|\mathbf{x}_{t-1} - \mathbf{g}\|_2 - \|\mathbf{x}_t - \mathbf{g}\|_2 + r_{\text{robot},t}, \tag{34}$$

and

$$c_t = \min\{1,\ \mathbf{1}(|y_t| > y_{\text{lim}}) + \mathbf{1}(\|\mathbf{v}_t\|_2 > v_{\text{lim}})\}, \tag{35}$$

where y_{lim} specifies the corridor boundary and v_{lim} is the velocity limit.

CostLQR. We use the CostLQR environment of Montenegro et al. (2024) with $d_S = d_A = 2$ and horizon $T = 50$. The dynamics are

$$s_{t+1} = As_t + Ba_t, \tag{36}$$

where $A \in \mathbb{R}^{d_S \times d_S}$ and $B \in \mathbb{R}^{d_S \times d_A}$. The reward and cost are defined as

$$r_t = -s_t^\top Q s_t, \qquad c_t = a_t^\top R a_t, \tag{37}$$

with

$$A = B = 0.9 I_2, \qquad R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.9 \end{bmatrix}, \qquad Q = \begin{bmatrix} 0.9 & 0 \\ 0 & 0.1 \end{bmatrix}. \tag{38}$$

Swimmer. We use `SafetySwimmerVelocity-v1` from Safety-Gymnasium (Ji et al., 2023), which is based on the MuJoCo Swimmer task (Todorov et al., 2012). We use $d_S = 8$, $d_A = 2$, horizon $T = 100$. The reward and cost are defined as

$$r_t = \dot{x}_t, \qquad c_t = \|a_t\|_2^2, \tag{39}$$

where $\dot{x}_t$ denotes the forward velocity computed from consecutive positions.

C.2 Threshold Settings

We consider single-, double-, and triple-threshold settings. Let $\mathcal{E}$ denote the set of available pre-shift thresholds. Within each comparison, all methods are given the same available thresholds.

For ZAPD, one converged pre-shift solution is independently trained for each $b \in \mathcal{E}$, using the same per-threshold interaction budget as in the single-threshold setting. Given a post-shift threshold $\hat{b}$, ZAPD applies the zero-shot update from the threshold in $\mathcal{E}$ nearest to $\hat{b}$; when two thresholds are equally distant, we use the larger one. CAPS constructs its policy library from the same available thresholds, while CCPO uses the same threshold set during its threshold-conditioned training.

Bullet Safety Gym. For all four Bullet Safety Gym tasks, we use

$$\mathcal{E}_{\text{single}} = \{40\}, \qquad \mathcal{E}_{\text{double}} = \{30, 50\}, \qquad \mathcal{E}_{\text{triple}} = \{20, 40, 60\}. \tag{40}$$

For immediate approximation error, we use

$$\mathcal{E}_{\text{eval}} = \{5, 10, \ldots, 75\}. \tag{41}$$

For first-convergence evaluation, we use only the single-threshold setting, with

$$\mathcal{E}_{\text{eval}} = \{5, 15, \ldots, 75\}. \tag{42}$$

To control for pre-shift environment interaction, ZAPD, CCPO, and CAPS use the same total interaction budget within each threshold setting. The trajectories used to construct ZAPD's gradient estimates are included in this budget.

Table 5: Matched pre-evaluation environment-interaction budgets (steps) for the Bullet Safety Gym tasks.

Environment	Single	Double	Triple
Ball-Circle	0.8M	1.6M	2.4M
Car-Circle	0.8M	1.6M	2.4M
Drone-Circle	1.2M	2.4M	3.6M
Drone-Run	1.2M	2.4M	3.6M

CostLQR and Swimmer. For CostLQR, we use

$$\mathcal{E}_{\text{single}} = \{7.5\}, \qquad \mathcal{E}_{\text{double}} = \{6.375, 8.625\}, \qquad \mathcal{E}_{\text{triple}} = \{6.375, 7.5, 8.625\}, \tag{43}$$

with

$$\mathcal{E}_{\text{eval}} = \{6.0, 6.375, 6.75, 7.125, 7.875, 8.25, 8.625, 9.0\}. \tag{44}$$

For Swimmer, we use

$$\mathcal{E}_{\text{single}} = \{2.5\}, \qquad \mathcal{E}_{\text{double}} = \{1.5, 3.5\}, \qquad \mathcal{E}_{\text{triple}} = \{1.5, 2.5, 3.5\}, \tag{45}$$

with

$$\mathcal{E}_{\text{eval}} = \{1.0, 1.5, 2.0, 3.0, 3.5, 4.0\}. \tag{46}$$

First-convergence experiments again use only the single-threshold setting.

To control for pre-shift environment interaction, ZAPD and CAPS use the same total interaction budget within each threshold setting. The trajectories used to construct ZAPD's gradient estimates are included in this budget.

Table 6: Matched pre-evaluation environment-interaction budgets (steps) for CostLQR and Swimmer.

Environment	Single	Double	Triple
CostLQR	10M	20M	30M
Swimmer	10M	20M	30M

The interaction budgets are matched within each base method, not across methods. Bullet Safety Gym uses off-policy CVPO, whereas CostLQR and Swimmer use on-policy C-PG and PPO-Lagrangian. The former reuses transitions through a critic and therefore needs fewer environment steps to reach the reference regime, even though the Bullet Safety Gym tasks are higher-dimensional.

C.3 Evaluation Metrics

For each post-shift threshold $\hat{b}$, we train five independent reference runs from scratch under $\hat{b}$ using the training durations specified in Appendix C.4. For each run, we average the reward and cost over the final 20% of training iterations, and then average these values across the five runs to obtain the reference values $\mathcal{R}(\hat{\theta}^*)$ and $\mathcal{J}(\hat{\theta}^*)$. Since all experiments in Section 5 use a single constraint ($m = 1$), we write $\mathcal{J}(\theta)$ for $\mathcal{J}_1(\theta)$ and omit the constraint index throughout this section.

Immediate Approximation Error. For the immediate-error evaluation, we apply each method once under the post-shift threshold $\hat{b}$ and then freeze the resulting policy. For ZAPD, we apply the zero-shot update specified in the corresponding main text experiment: Section 5.2 uses the primal-only update, while Section 5.3 evaluates the primal-only and full primal-dual variants. For CAPS, we perform one filter-and-select operation followed by one primal-dual update. For CCPO, we condition the actor on $\hat{b}$. We then average the frozen policy for 50 episodes without further parameter updates.

Let $\mathcal{E}_{\text{eval}}$ denote the set of evaluated post-shift thresholds for a given environment and setting, with $n = |\mathcal{E}_{\text{eval}}|$, and let $s = 1, \dots, S$ index the $S = 20$ independent runs, each with its own random seed. For $q \in \{\mathcal{R}, \mathcal{J}\}$, let $x_s^q(\hat{b})$ denote the corresponding reward or cost estimate for run s, averaged over the 50 episodes, and let $x^{*,q}(\hat{b})$ denote the corresponding reference value. We first compute the relative error for each run:

$$e_s^q(\hat{b}) = \frac{\left| x_s^q(\hat{b}) - x^{*,q}(\hat{b}) \right|}{\left| x^{*,q}(\hat{b}) \right|} \times 100. \tag{47}$$

We then average across the evaluated thresholds and across reward and cost within each run:

$$E_s = \frac{1}{2n} \sum_{\hat{b} \in \mathcal{E}_{\text{eval}}} \left(e_s^{\mathcal{R}}(\hat{b}) + e_s^{\mathcal{J}}(\hat{b}) \right). \tag{48}$$

The immediate approximation error reported in the main tables is the mean and sample standard deviation of these run-level scores:

$$\mu = \frac{1}{S} \sum_{s=1}^{S} E_s, \qquad \sigma = \sqrt{\frac{1}{S-1} \sum_{s=1}^{S} (E_s - \mu)^2}. \tag{49}$$

First-Convergence Iteration. After adaptation, training resumes under the post-shift threshold $\hat{b}$. For post-shift run i, let $\mathcal{R}_i^w(k)$ and $\mathcal{J}_i^w(k)$ denote the reward and cost averaged over a window of w iterations ending at iteration k. We define

$$z_{\mathcal{R},i}^w(k) = \frac{\mathcal{R}_i^w(k) - \mathcal{R}(\hat{\theta}^*)}{|\mathcal{R}(\hat{\theta}^*)|}, \qquad z_{\mathcal{J},i}^w(k) = \frac{\mathcal{J}_i^w(k) - \mathcal{J}(\hat{\theta}^*)}{|\mathcal{J}(\hat{\theta}^*)|}. \tag{50}$$

The first-convergence iteration is

$$k_i^{\text{conv},w} = \min \left\{ k \geq w : |z_{\mathcal{R},i}^w(k)| \leq \varepsilon \ \wedge \ |z_{\mathcal{J},i}^w(k)| \leq \varepsilon \right\}, \tag{51}$$

where $\varepsilon = 0.05$ and $w = 1$. We use $w = 1$ consistently with the evaluation protocol of CCPO (Yao et al., 2023), where each policy evaluation is already averaged over 50 episodes.

For each threshold shift, P_{50} and P_{80} are the 50th and 80th percentiles of $\{k_i^{\text{conv},w}\}_{i=1}^{20}$, respectively. If the required fraction of runs has not converged within the evaluation horizon, the corresponding percentile is set to the maximum evaluation iteration for that environment. For Ball-Circle, Car-Circle, Drone-Circle, and Drone-Run, these maximum iterations are 80, 80, 120, and 120, respectively; for CostLQR and Swimmer, the maximum is $1{,}000$.

Finally, for $q \in \{50, 80\}$, we report

$$\overline{\mathrm{P}}_q = \frac{1}{|\mathcal{E}_{\text{eval}}|} \sum_{\hat{b} \in \mathcal{E}_{\text{eval}}} \mathrm{P}_q(\hat{b}). \tag{52}$$

C.4 Implementation Details

ZAPD. The primal update of ZAPD is applied only to the weights and bias of the policy-mean network, leaving the action standard deviation at its pre-shift values.

For the Bullet Safety Gym experiments, pre-shift training uses 80, 80, 120, and 120 iterations for Ball-Circle, Car-Circle, Drone-Circle, and Drone-Run, respectively. For CostLQR and Swimmer, pre-shift training uses 2000 iterations.

The EMA coefficient for the reward and constraint-gradient estimates is $\beta = 0.95$ for the Bullet Safety Gym tasks and $\beta = 0.99$ for CostLQR and Swimmer. We start the EMA updates at $K_{\mathrm{EMA}} = 30$, 30, 40, and 30 for Ball-Circle, Car-Circle, Drone-Circle, and Drone-Run, respectively, and at $K_{\mathrm{EMA}} = 1200$ for CostLQR and Swimmer. For the practical checks corresponding to Assumptions 4 and 5 in Algorithm 1, we use quantities estimated at the pre-shift iterate θ_{k_c}. For Assumption 4, we treat constraint i as active if $|\mathcal{J}_i(\theta_{k_c}) - b_i| \leq 0.1 b_i$. We use the same relative tolerance for all environments and do not tune it per environment. For Assumption 5, we require the smallest singular value of the estimated constraint Jacobian $\widetilde{\nabla}_\theta \boldsymbol{\mathcal{J}}(\theta_{k_c})$ to exceed 0.001. In the single-constraint case ($m = 1$), which covers all main text experiments, this reduces to $\|\widetilde{\nabla}_\theta \mathcal{J}(\theta_{k_c})\|_2 > 0.001$.

For the AlgaeDICE critics used only by the dual ZAPD update, we use regularization coefficient $\alpha = 0.01$, discount factor $\gamma = 0.99$, target-critic EMA coefficient 5×10^{-3}, critic learning rate 10^{-3}, and minibatch size 256. We perform 20 critic updates during each normal pre-shift phase and 500 additional updates before the zero-shot dual correction.

CCPO. We implement CCPO (Yao et al., 2023) based on the open-source CVPO implementation (Liu et al., 2022) as the base. Both the policy and reward/cost action-value functions are conditioned on the constraint threshold. Training first collects trajectories from behavior policies trained at the available thresholds and then jointly updates the threshold-conditioned VVE critics and CVI policy; the critics remain trainable throughout this stage. We use two-layer $[256, 256]$ ReLU policy and Q networks, a fixed action standard deviation of 0.5, $\gamma = 0.97$, VVE critic learning rate 5×10^{-4}, minibatch size 256, 16 action particles, and one M-step update. The E-step KL threshold is 0.1, the mean and covariance M-step KL thresholds are 10^{-2} and 5×10^{-3}, respectively, and the E-step dual problem is solved with L-BFGS-B.

CAPS. CAPS (Chemingui et al., 2025) maintains a set of policies spanning the reward–cost trade-off, from reward-oriented to cost-oriented policies, and selects among them according to the deployment constraint. In its original setting, this policy set is constructed using a grid of scalarization coefficients λ between the reward-only and cost-only policies.

In our primal-dual setting, training policies under different constraint thresholds b_j produces policies associated with different optimal multipliers $\lambda^*(b_j)$, thereby providing an analogous set of policies along the reward–cost trade-off. We therefore preserve the policy-switching principle of CAPS, while performing filtering and selection at the policy level.

For each available threshold b_j, we train a policy $\pi^*_{b_j}$ and record its converged reward and cost $\mathcal{R}^*(b_j)$ and $\mathcal{J}^*(b_j)$. After observing $\hat{b}$, we form

$$\mathcal{S}(\hat{b}) = \left\{ j : \mathcal{J}^*(b_j) \leq \hat{b} \right\}. \tag{53}$$

If $\mathcal{S}(\hat{b})$ is nonempty, CAPS selects

$$j^* = \arg\max_{j \in \mathcal{S}(\hat{b})} \mathcal{R}^*(b_j). \tag{54}$$

Otherwise, it selects the library policy with the smallest converged cost. The selection itself requires no post-shift environment interaction or retraining.

C-PG and PPO-Lagrangian. We use C-PG from Montenegro et al. (2024) and PPO-Lagrangian (PPO-L) using OmniSafe (Ji et al., 2024). Both use $5{,}000$ environment steps per iteration, reward and cost discount factors of 0.99, actor learning rate 10^{-3}, and dual learning rate 10^{-2} with Adam. C-PG performs one Monte-Carlo policy-gradient update using GPOMDP reward and cost gradients. PPO-L uses two actor updates per rollout batch, clipping ratio 0.2, GAE parameter 1.0, target KL 0.02 with KL-based early stopping, and gradient-norm clipping at 40.

For both the linear and NN policies, the action distribution is Gaussian with fixed diagonal variance $\sigma^2 = 10^{-3}$. The linear policy uses mean $\mu_\theta(s) = Ws$, where $W \in \mathbb{R}^{d_A \times d_S}$ and there is no bias. The NN policy uses a two-layer $[32, 32]$ MLP with $\tanh$ activations. All networks use Kaiming-uniform initialization.

D Detailed Results

D.1 Breakdowns of Table 1

In this section, we provide the immediate approximation errors for different post-shift thresholds $\hat{b}$ in the Bullet Safety Gym tasks. We also present the breakdown between reward error and cost error for the single-, double-, and triple-threshold settings.

Table 7: Detailed immediate approximation error (%) for **Ball-Circle**. Each cell reports reward error (mean ± standard deviation) / cost error (mean ± standard deviation) over 20 runs.

		Ball-Circle		
Setting	$\hat{b}$	ZAPD	CCPO	CAPS
Single	5	30.37 ± 4.21 / 94.22 ± 83.83	96.43 ± 3.57 / 99.26 ± 2.34	13.38 ± 0.92 / 628.68 ± 35.55
	10	26.49 ± 3.97 / 59.78 ± 38.46	95.48 ± 3.98 / 99.23 ± 2.49	11.00 ± 0.90 / 293.08 ± 19.18
	15	21.87 ± 3.60 / 46.60 ± 29.62	94.06 ± 5.50 / 99.85 ± 0.54	9.32 ± 0.89 / 167.11 ± 13.03
	20	16.80 ± 3.16 / 33.88 ± 22.52	91.46 ± 7.17 / 99.91 ± 0.38	7.37 ± 0.87 / 104.78 ± 9.99
	25	12.21 ± 2.61 / 23.27 ± 19.70	87.49 ± 9.32 / 99.74 ± 0.95	5.05 ± 0.86 / 63.83 ± 7.99
	30	6.49 ± 2.25 / 14.97 ± 14.34	75.74 ± 12.97 / 99.59 ± 1.18	3.56 ± 0.84 / 39.01 ± 6.78
	35	1.99 ± 1.22 / 8.66 ± 8.85	54.51 ± 13.93 / 99.53 ± 1.21	1.62 ± 0.83 / 18.05 ± 5.76
	40	0.71 ± 0.54 / 4.42 ± 3.01	6.78 ± 3.04 / 8.38 ± 6.06	0.66 ± 0.46 / 5.33 ± 3.22
	45	2.53 ± 0.87 / 8.52 ± 6.30	2.35 ± 2.27 / 76.62 ± 7.15	1.70 ± 0.81 / 6.41 ± 4.32
	50	6.82 ± 1.38 / 13.61 ± 5.85	6.95 ± 3.73 / 74.96 ± 6.64	3.40 ± 0.80 / 15.46 ± 3.99
	55	11.49 ± 1.67 / 13.29 ± 6.17	10.59 ± 4.20 / 63.87 ± 6.92	4.98 ± 0.78 / 23.27 ± 3.63
	60	15.89 ± 1.98 / 11.41 ± 5.87	12.91 ± 4.49 / 54.19 ± 7.21	6.35 ± 0.77 / 29.30 ± 3.34
	65	20.37 ± 2.25 / 9.29 ± 5.48	15.59 ± 4.82 / 43.28 ± 7.62	8.30 ± 0.76 / 34.71 ± 3.09
	70	23.81 ± 2.45 / 5.64 ± 4.85	16.77 ± 4.98 / 31.46 ± 7.18	9.24 ± 0.75 / 40.81 ± 2.80
	75	26.40 ± 2.66 / 8.05 ± 6.27	17.13 ± 5.19 / 30.49 ± 7.12	9.30 ± 0.75 / 41.46 ± 2.77
Double	5	29.31 ± 4.64 / 68.75 ± 25.71	90.53 ± 9.60 / 634.09 ± 655.24	10.02 ± 0.73 / 477.27 ± 27.26
	10	23.86 ± 4.18 / 59.76 ± 28.42	86.89 ± 13.93 / 322.20 ± 376.75	7.72 ± 0.72 / 211.40 ± 14.70
	15	17.24 ± 3.74 / 46.73 ± 30.20	70.61 ± 28.37 / 223.63 ± 248.92	6.09 ± 0.71 / 111.61 ± 9.99
	20	10.18 ± 2.95 / 31.20 ± 21.88	51.16 ± 32.67 / 105.40 ± 92.28	4.19 ± 0.69 / 62.23 ± 7.66
	25	4.02 ± 1.78 / 16.16 ± 13.44	18.51 ± 16.05 / 78.23 ± 20.17	1.95 ± 0.68 / 29.79 ± 6.13
	30	0.66 ± 0.43 / 5.52 ± 4.14	4.48 ± 0.88 / 11.87 ± 7.47	0.65 ± 0.58 / 10.41 ± 5.07
	35	1.54 ± 1.07 / 27.75 ± 14.49	2.68 ± 1.41 / 12.26 ± 7.52	1.27 ± 0.63 / 5.83 ± 3.44
	40	5.13 ± 2.43 / 7.23 ± 6.82	3.32 ± 1.70 / 9.18 ± 9.65	2.80 ± 0.67 / 16.93 ± 3.73
	45	1.42 ± 0.91 / 4.33 ± 3.62	3.64 ± 1.68 / 7.11 ± 8.14	4.41 ± 0.89 / 24.88 ± 3.98
	50	0.60 ± 0.32 / 3.17 ± 1.71	1.74 ± 0.70 / 4.18 ± 2.42	0.57 ± 0.38 / 4.16 ± 3.26
	55	2.37 ± 1.09 / 3.87 ± 2.37	0.88 ± 0.87 / 12.05 ± 6.32	1.65 ± 0.67 / 6.52 ± 3.70
	60	6.62 ± 1.50 / 3.63 ± 3.03	0.92 ± 0.63 / 9.77 ± 4.86	3.06 ± 0.66 / 13.70 ± 3.71
	65	11.99 ± 1.80 / 4.01 ± 2.75	2.15 ± 1.63 / 3.74 ± 3.25	5.08 ± 0.65 / 20.30 ± 3.43
	70	15.60 ± 2.22 / 4.82 ± 4.07	3.70 ± 2.25 / 6.04 ± 3.38	6.06 ± 0.64 / 27.74 ± 3.11
	75	18.64 ± 2.36 / 4.15 ± 2.52	4.51 ± 2.73 / 6.95 ± 3.82	6.11 ± 0.64 / 28.53 ± 3.07
Triple	5	24.74 ± 3.79 / 70.84 ± 23.16	71.22 ± 25.64 / 370.96 ± 619.97	6.64 ± 1.10 / 327.87 ± 55.93
	10	16.13 ± 2.68 / 57.31 ± 18.82	49.52 ± 28.75 / 204.97 ± 359.36	4.41 ± 1.08 / 130.81 ± 30.17
	15	6.48 ± 1.39 / 36.14 ± 17.17	15.59 ± 7.10 / 91.34 ± 6.03	2.83 ± 1.06 / 56.85 ± 20.50
	20	0.63 ± 0.46 / 6.84 ± 4.94	3.31 ± 0.82 / 23.28 ± 7.92	1.12 ± 0.93 / 20.84 ± 15.47
	25	1.87 ± 1.12 / 77.60 ± 23.38	2.64 ± 1.80 / 32.84 ± 31.40	1.27 ± 0.75 / 10.12 ± 7.92
	30	6.25 ± 2.15 / 14.48 ± 10.42	3.06 ± 1.82 / 30.28 ± 28.47	2.44 ± 1.05 / 17.37 ± 9.32
	35	2.12 ± 1.37 / 8.85 ± 8.04	2.12 ± 1.23 / 12.95 ± 11.40	4.27 ± 1.03 / 29.01 ± 9.34
	40	0.65 ± 0.42 / 3.99 ± 3.26	3.18 ± 0.97 / 6.60 ± 4.48	0.68 ± 0.45 / 5.45 ± 3.21
	45	2.47 ± 0.97 / 9.81 ± 6.64	5.33 ± 2.66 / 20.42 ± 16.14	1.70 ± 0.82 / 6.36 ± 4.32
	50	4.26 ± 1.99 / 5.32 ± 4.71	6.58 ± 4.30 / 27.89 ± 24.76	3.40 ± 0.80 / 15.40 ± 4.02
	55	1.13 ± 0.77 / 2.88 ± 2.46	4.84 ± 4.72 / 18.71 ± 21.66	1.48 ± 0.65 / 10.08 ± 2.32
	60	0.47 ± 0.40 / 1.79 ± 1.46	0.49 ± 0.46 / 3.45 ± 1.67	0.54 ± 0.32 / 2.19 ± 1.40
	65	2.79 ± 0.70 / 3.58 ± 2.44	0.96 ± 0.60 / 2.22 ± 1.80	2.03 ± 0.62 / 6.16 ± 1.99
	70	6.35 ± 0.78 / 8.68 ± 2.50	2.05 ± 1.02 / 9.04 ± 3.10	3.04 ± 0.62 / 14.92 ± 1.81
	75	9.76 ± 1.29 / 6.56 ± 3.36	2.51 ± 1.56 / 9.80 ± 3.85	3.10 ± 0.62 / 15.86 ± 1.79

Table 8: Detailed immediate approximation error (%) for **Car-Circle**. Each cell reports reward error (mean ± standard deviation) / cost error (mean ± standard deviation) over 20 runs.

		Car-Circle		
Setting	$\hat{b}$	ZAPD	CCPO	CAPS
Single	5	60.30 ± 9.70 / 99.84 ± 0.49	96.00 ± 3.94 / 238.22 ± 293.11	5.85 ± 1.39 / 553.32 ± 72.86
	10	54.60 ± 9.09 / 99.73 ± 0.87	92.03 ± 6.91 / 120.89 ± 88.47	4.43 ± 1.29 / 302.69 ± 43.40
	15	47.35 ± 6.96 / 97.51 ± 6.33	85.68 ± 10.61 / 110.29 ± 49.85	4.44 ± 1.30 / 183.92 ± 30.50
	20	39.09 ± 5.22 / 85.84 ± 25.37	73.63 ± 16.42 / 100.28 ± 14.58	4.02 ± 1.27 / 128.08 ± 24.69
	25	30.01 ± 4.01 / 60.93 ± 27.16	52.71 ± 17.56 / 98.94 ± 3.08	2.81 ± 1.06 / 75.94 ± 18.78
	30	19.65 ± 2.90 / 23.55 ± 13.32	29.41 ± 9.37 / 99.71 ± 0.47	1.99 ± 0.98 / 51.31 ± 16.50
	35	8.35 ± 1.63 / 11.18 ± 8.52	14.04 ± 2.05 / 98.79 ± 1.28	1.13 ± 0.81 / 29.18 ± 13.78
	40	0.84 ± 0.48 / 6.30 ± 3.95	5.70 ± 1.09 / 18.89 ± 14.51	0.95 ± 0.96 / 16.37 ± 10.59
	45	2.80 ± 1.15 / 67.75 ± 22.85	1.26 ± 0.79 / 108.99 ± 8.05	1.40 ± 1.11 / 10.06 ± 6.67
	50	9.04 ± 1.11 / 89.49 ± 17.58	1.37 ± 0.94 / 111.12 ± 9.28	3.06 ± 1.23 / 8.45 ± 6.75
	55	14.00 ± 1.35 / 89.12 ± 14.13	2.74 ± 2.18 / 105.00 ± 7.23	3.68 ± 1.19 / 13.75 ± 8.71
	60	18.16 ± 1.53 / 85.83 ± 9.91	6.43 ± 3.10 / 96.40 ± 8.31	4.23 ± 1.24 / 20.73 ± 8.35
	65	21.92 ± 2.10 / 79.85 ± 10.03	9.24 ± 2.87 / 85.65 ± 4.81	4.60 ± 1.22 / 26.32 ± 7.77
	70	26.22 ± 2.28 / 73.21 ± 9.89	12.68 ± 3.02 / 76.74 ± 5.73	5.84 ± 1.24 / 32.69 ± 7.40
	75	30.25 ± 2.71 / 67.80 ± 8.23	14.78 ± 2.86 / 68.56 ± 5.98	6.59 ± 1.20 / 36.51 ± 6.92
Double	5	58.49 ± 7.34 / 99.78 ± 0.68	46.66 ± 33.30 / 97.83 ± 32.70	3.88 ± 1.08 / 477.40 ± 74.65
	10	50.28 ± 6.95 / 99.81 ± 0.66	33.02 ± 29.92 / 100.65 ± 29.92	2.47 ± 1.12 / 255.12 ± 47.20
	15	39.16 ± 3.52 / 99.43 ± 1.64	23.33 ± 27.80 / 91.31 ± 17.36	2.57 ± 1.09 / 152.00 ± 31.06
	20	26.80 ± 2.88 / 82.20 ± 22.17	9.48 ± 13.24 / 81.23 ± 26.23	2.09 ± 1.07 / 101.77 ± 25.42
	25	12.35 ± 1.74 / 38.83 ± 23.32	3.42 ± 1.90 / 65.72 ± 27.36	1.01 ± 0.96 / 55.88 ± 18.96
	30	0.93 ± 0.59 / 18.15 ± 11.10	2.86 ± 0.44 / 35.16 ± 12.01	0.92 ± 0.60 / 34.29 ± 17.70
	35	3.70 ± 1.40 / 125.90 ± 22.61	3.47 ± 0.81 / 29.31 ± 18.53	1.47 ± 0.80 / 16.62 ± 13.67
	40	14.96 ± 3.39 / 7.50 ± 5.68	4.46 ± 1.07 / 26.25 ± 22.08	2.28 ± 1.06 / 11.12 ± 8.58
	45	5.80 ± 1.86 / 5.16 ± 3.58	4.41 ± 0.98 / 18.01 ± 17.51	1.76 ± 0.57 / 17.49 ± 10.13
	50	0.69 ± 0.56 / 4.99 ± 4.68	4.58 ± 0.76 / 9.97 ± 9.45	0.43 ± 0.34 / 8.98 ± 8.42
	55	3.09 ± 0.87 / 32.36 ± 14.13	4.05 ± 1.85 / 34.79 ± 23.46	0.70 ± 0.50 / 7.01 ± 6.14
	60	7.86 ± 1.43 / 55.47 ± 10.31	5.11 ± 4.07 / 45.95 ± 24.39	1.25 ± 0.54 / 12.25 ± 5.77
	65	12.08 ± 1.72 / 57.02 ± 6.46	7.95 ± 8.29 / 50.45 ± 32.58	1.63 ± 0.55 / 17.89 ± 6.42
	70	16.59 ± 1.80 / 52.73 ± 6.56	12.65 ± 12.19 / 53.15 ± 39.87	2.98 ± 0.52 / 24.23 ± 6.94
	75	19.93 ± 2.10 / 51.14 ± 6.48	17.88 ± 16.80 / 58.06 ± 51.40	3.66 ± 0.54 / 28.53 ± 6.53
Triple	5	39.04 ± 5.25 / 98.99 ± 2.60	20.74 ± 21.83 / 91.91 ± 13.42	2.07 ± 0.89 / 317.92 ± 101.97
	10	27.36 ± 4.11 / 97.11 ± 8.91	8.03 ± 4.50 / 91.01 ± 12.15	0.76 ± 0.81 / 151.99 ± 62.67
	15	12.53 ± 2.65 / 85.69 ± 15.70	3.03 ± 1.86 / 85.16 ± 13.29	0.80 ± 0.78 / 80.44 ± 44.58
	20	1.30 ± 3.86 / 27.60 ± 58.58	1.03 ± 0.77 / 59.27 ± 18.71	0.59 ± 0.57 / 46.73 ± 32.45
	25	4.70 ± 7.58 / 245.79 ± 61.39	1.16 ± 0.52 / 39.34 ± 25.81	1.06 ± 0.51 / 24.10 ± 18.06
	30	19.15 ± 3.12 / 19.75 ± 15.45	1.55 ± 0.73 / 26.49 ± 25.19	1.80 ± 0.73 / 19.91 ± 11.38
	35	8.48 ± 1.75 / 12.83 ± 8.90	2.23 ± 0.73 / 15.73 ± 14.82	2.61 ± 1.04 / 25.71 ± 15.78
	40	0.90 ± 0.73 / 6.35 ± 4.75	3.28 ± 0.79 / 9.05 ± 5.28	0.97 ± 0.94 / 14.77 ± 10.05
	45	3.58 ± 1.34 / 69.60 ± 15.91	3.51 ± 0.75 / 11.67 ± 9.41	1.43 ± 1.15 / 10.38 ± 5.44
	50	14.05 ± 3.89 / 8.35 ± 6.07	4.63 ± 0.99 / 13.87 ± 10.77	3.10 ± 1.28 / 9.10 ± 7.11
	55	6.04 ± 2.15 / 6.75 ± 3.89	4.13 ± 1.14 / 11.79 ± 6.24	0.88 ± 0.52 / 15.92 ± 8.86
	60	0.54 ± 0.53 / 3.29 ± 3.24	2.36 ± 0.85 / 23.71 ± 7.18	0.77 ± 0.69 / 7.17 ± 7.27
	65	2.56 ± 1.11 / 13.39 ± 8.75	1.37 ± 0.80 / 32.26 ± 8.16	0.95 ± 0.82 / 6.28 ± 3.38
	70	7.43 ± 1.73 / 22.75 ± 9.58	2.05 ± 1.04 / 34.38 ± 9.31	2.03 ± 0.99 / 9.74 ± 5.65
	75	12.00 ± 2.06 / 27.81 ± 7.49	3.14 ± 1.85 / 33.08 ± 8.81	2.78 ± 0.99 / 13.97 ± 6.34

Table 9: Detailed immediate approximation error (%) for **Drone-Circle**. Each cell reports reward error (mean ± standard deviation) / cost error (mean ± standard deviation) over 20 runs.

		Drone-Circle		
Setting	$\hat{b}$	ZAPD	CCPO	CAPS
Single	5	58.17 ± 11.20 / 46.19 ± 47.09	97.62 ± 1.34 / 90.75 ± 9.63	14.49 ± 3.91 / 494.59 ± 30.58
	10	52.52 ± 10.94 / 25.88 ± 17.34	96.11 ± 2.56 / 94.41 ± 7.25	9.56 ± 3.61 / 240.05 ± 26.74
	15	45.83 ± 11.91 / 33.22 ± 19.21	93.31 ± 4.82 / 97.67 ± 2.04	7.53 ± 3.17 / 137.40 ± 17.75
	20	38.67 ± 9.67 / 30.85 ± 19.09	87.57 ± 7.45 / 97.65 ± 2.23	6.92 ± 4.31 / 90.09 ± 11.32
	25	27.20 ± 7.76 / 28.06 ± 15.59	75.36 ± 11.13 / 97.14 ± 2.59	6.76 ± 3.43 / 57.41 ± 12.42
	30	13.61 ± 7.24 / 16.35 ± 11.93	52.95 ± 16.70 / 96.71 ± 3.29	5.28 ± 3.20 / 31.75 ± 8.45
	35	6.05 ± 4.17 / 8.78 ± 9.74	24.90 ± 11.35 / 96.04 ± 3.57	6.69 ± 4.06 / 20.57 ± 7.63
	40	4.07 ± 2.45 / 4.40 ± 3.82	10.48 ± 5.43 / 11.14 ± 6.47	4.18 ± 2.70 / 7.63 ± 4.85
	45	4.39 ± 2.65 / 14.37 ± 6.98	9.17 ± 6.15 / 87.78 ± 14.28	3.97 ± 2.49 / 7.67 ± 5.25
	50	4.92 ± 4.75 / 28.55 ± 15.33	15.85 ± 11.46 / 99.23 ± 22.19	3.16 ± 2.43 / 13.84 ± 5.06
	55	7.63 ± 5.64 / 32.52 ± 19.70	25.01 ± 15.50 / 89.20 ± 29.44	3.56 ± 2.87 / 22.29 ± 5.91
	60	13.21 ± 6.67 / 35.55 ± 18.23	28.50 ± 14.14 / 88.38 ± 22.13	4.00 ± 3.44 / 27.76 ± 5.35
	65	19.06 ± 7.07 / 34.15 ± 19.70	33.62 ± 13.85 / 77.39 ± 24.79	4.23 ± 3.38 / 33.50 ± 5.34
	70	25.75 ± 9.57 / 34.35 ± 20.83	39.30 ± 14.69 / 64.82 ± 25.02	3.69 ± 3.29 / 36.66 ± 4.35
	75	30.00 ± 9.09 / 32.50 ± 18.45	40.29 ± 15.74 / 58.40 ± 27.90	3.93 ± 3.51 / 40.70 ± 4.01
Double	5	51.36 ± 14.36 / 48.63 ± 60.55	87.71 ± 15.28 / 72.62 ± 25.69	10.54 ± 5.61 / 373.58 ± 57.83
	10	45.28 ± 11.63 / 48.64 ± 23.89	82.67 ± 16.43 / 84.73 ± 16.45	7.57 ± 4.28 / 170.16 ± 33.39
	15	35.20 ± 9.97 / 55.10 ± 22.18	72.64 ± 20.79 / 85.79 ± 10.44	6.12 ± 3.52 / 88.68 ± 22.70
	20	21.98 ± 6.80 / 55.02 ± 22.82	50.01 ± 20.29 / 86.43 ± 12.08	5.84 ± 3.59 / 54.69 ± 18.70
	25	8.28 ± 4.34 / 41.54 ± 17.80	19.27 ± 11.11 / 89.07 ± 8.09	6.25 ± 3.76 / 28.09 ± 15.65
	30	3.46 ± 2.13 / 6.88 ± 5.18	5.50 ± 3.95 / 30.65 ± 12.56	4.99 ± 3.67 / 12.06 ± 8.30
	35	4.32 ± 2.76 / 68.80 ± 19.98	5.11 ± 4.12 / 16.61 ± 12.03	4.56 ± 4.07 / 8.64 ± 4.99
	40	13.68 ± 8.19 / 6.20 ± 4.63	4.87 ± 3.31 / 14.30 ± 9.62	4.21 ± 4.15 / 14.19 ± 8.41
	45	5.89 ± 5.06 / 5.17 ± 4.59	4.72 ± 3.89 / 8.33 ± 4.35	4.72 ± 3.51 / 12.65 ± 8.22
	50	3.00 ± 2.23 / 3.02 ± 2.14	4.75 ± 4.00 / 9.64 ± 4.95	5.39 ± 3.98 / 5.68 ± 4.26
	55	4.17 ± 2.10 / 7.92 ± 6.17	5.29 ± 4.30 / 9.52 ± 9.54	4.65 ± 2.89 / 8.14 ± 3.62
	60	4.77 ± 3.48 / 18.84 ± 12.10	7.21 ± 5.64 / 13.55 ± 13.60	4.41 ± 2.57 / 15.16 ± 5.37
	65	5.65 ± 5.15 / 18.39 ± 12.42	8.08 ± 6.61 / 18.12 ± 12.36	4.58 ± 2.61 / 21.95 ± 5.18
	70	7.80 ± 7.19 / 21.17 ± 13.55	10.24 ± 6.55 / 17.72 ± 13.29	4.96 ± 3.65 / 26.66 ± 5.21
	75	13.88 ± 11.10 / 19.84 ± 14.19	12.51 ± 9.55 / 19.45 ± 13.13	4.92 ± 3.84 / 31.14 ± 5.39
Triple	5	33.03 ± 10.61 / 38.77 ± 25.99	71.62 ± 20.04 / 67.10 ± 27.84	8.76 ± 4.26 / 250.17 ± 62.48
	10	22.02 ± 7.89 / 60.47 ± 18.44	52.38 ± 23.96 / 79.84 ± 14.99	5.39 ± 2.76 / 101.66 ± 33.22
	15	9.87 ± 4.97 / 63.97 ± 15.06	23.65 ± 15.14 / 88.11 ± 9.84	4.63 ± 2.83 / 45.73 ± 23.63
	20	4.52 ± 3.14 / 12.24 ± 9.87	6.97 ± 5.74 / 45.50 ± 10.90	3.62 ± 3.56 / 19.05 ± 17.88
	25	4.83 ± 3.45 / 149.70 ± 25.87	4.85 ± 4.09 / 38.95 ± 35.62	3.62 ± 2.63 / 13.23 ± 10.60
	30	14.91 ± 7.57 / 18.84 ± 11.50	5.95 ± 3.61 / 26.33 ± 25.66	3.87 ± 2.34 / 20.82 ± 10.57
	35	3.59 ± 2.99 / 9.51 ± 6.93	4.71 ± 2.77 / 13.29 ± 9.19	3.63 ± 2.30 / 27.13 ± 9.92
	40	3.56 ± 2.59 / 5.95 ± 4.52	2.97 ± 2.18 / 7.80 ± 5.45	5.56 ± 3.56 / 11.29 ± 9.49
	45	3.66 ± 2.35 / 16.83 ± 7.99	3.65 ± 2.89 / 7.54 ± 5.84	4.17 ± 3.26 / 9.09 ± 10.91
	50	8.62 ± 5.52 / 5.61 ± 3.97	4.43 ± 2.78 / 5.54 ± 5.28	4.22 ± 2.95 / 14.76 ± 11.72
	55	4.72 ± 3.17 / 4.65 ± 3.05	4.93 ± 3.50 / 6.08 ± 5.20	4.58 ± 2.98 / 11.98 ± 6.56
	60	2.86 ± 2.37 / 5.19 ± 3.98	4.41 ± 2.79 / 7.12 ± 4.73	3.65 ± 3.07 / 6.04 ± 6.88
	65	3.70 ± 2.65 / 6.96 ± 4.74	5.07 ± 3.38 / 9.31 ± 5.81	4.42 ± 3.52 / 8.70 ± 7.87
	70	3.68 ± 1.97 / 13.24 ± 6.42	7.23 ± 4.65 / 10.90 ± 7.64	4.08 ± 3.34 / 12.09 ± 7.70
	75	3.77 ± 2.18 / 15.02 ± 8.06	7.83 ± 6.87 / 13.54 ± 8.85	5.17 ± 4.48 / 18.00 ± 7.45

Table 10: Detailed immediate approximation error (%) for **Drone-Run**. Each cell reports reward error (mean ± standard deviation) / cost error (mean ± standard deviation) over 20 runs.

		Drone-Run		
Setting	$\hat{b}$	ZAPD	CCPO	CAPS
Single	5	11.58 ± 11.73 / 301.78 ± 266.40	61.08 ± 23.38 / 266.41 ± 439.71	9.13 ± 12.90 / 290.03 ± 317.34
	10	9.99 ± 11.75 / 125.26 ± 133.40	51.14 ± 25.45 / 176.62 ± 230.18	9.09 ± 12.58 / 143.40 ± 159.27
	15	11.09 ± 11.74 / 83.67 ± 75.40	42.65 ± 31.74 / 119.82 ± 115.42	9.11 ± 12.83 / 88.14 ± 93.77
	20	10.26 ± 10.68 / 81.63 ± 64.17	28.01 ± 15.37 / 98.23 ± 7.39	8.36 ± 12.00 / 79.32 ± 78.72
	25	9.09 ± 9.47 / 56.42 ± 37.07	17.40 ± 7.35 / 100.00 ± 0.00	8.79 ± 12.28 / 70.68 ± 55.09
	30	9.04 ± 9.06 / 66.79 ± 51.26	15.03 ± 5.32 / 99.81 ± 0.85	8.65 ± 10.15 / 72.57 ± 57.88
	35	8.65 ± 8.12 / 55.39 ± 34.91	12.03 ± 2.55 / 99.09 ± 2.99	8.94 ± 9.12 / 65.03 ± 44.52
	40	8.58 ± 4.97 / 54.56 ± 24.83	9.82 ± 1.68 / 39.49 ± 27.23	9.71 ± 7.91 / 64.34 ± 45.36
	45	7.81 ± 5.85 / 57.51 ± 40.92	10.38 ± 10.02 / 355.45 ± 56.55	9.55 ± 8.14 / 67.37 ± 48.85
	50	8.82 ± 6.58 / 65.60 ± 68.67	10.95 ± 11.16 / 299.18 ± 38.17	9.67 ± 7.92 / 60.07 ± 40.64
	55	8.35 ± 5.75 / 63.81 ± 71.37	16.24 ± 13.69 / 282.52 ± 65.88	9.53 ± 8.09 / 61.42 ± 40.57
	60	7.89 ± 4.76 / 56.22 ± 60.95	17.47 ± 16.87 / 230.00 ± 72.79	10.61 ± 7.11 / 55.80 ± 38.66
	65	12.50 ± 5.64 / 59.79 ± 42.47	21.00 ± 18.52 / 140.65 ± 63.66	13.73 ± 6.63 / 58.26 ± 30.84
	70	14.71 ± 7.61 / 54.55 ± 40.03	22.45 ± 22.04 / 106.64 ± 53.90	17.57 ± 7.13 / 60.97 ± 28.39
	75	13.55 ± 6.75 / 59.21 ± 36.18	23.93 ± 24.51 / 103.27 ± 52.98	15.97 ± 6.85 / 60.83 ± 28.24
Double	5	6.01 ± 11.74 / 213.76 ± 193.60	37.66 ± 29.43 / 543.61 ± 512.47	6.09 ± 7.72 / 278.14 ± 217.50
	10	5.26 ± 8.97 / 89.16 ± 78.01	28.80 ± 25.97 / 239.53 ± 232.69	6.01 ± 7.77 / 140.16 ± 98.73
	15	6.00 ± 8.34 / 74.41 ± 62.37	22.23 ± 24.19 / 130.51 ± 106.65	5.98 ± 7.73 / 79.90 ± 48.26
	20	4.88 ± 4.91 / 57.98 ± 35.06	16.13 ± 19.03 / 127.65 ± 110.64	5.39 ± 6.90 / 68.50 ± 42.90
	25	3.69 ± 4.53 / 51.29 ± 31.51	7.08 ± 4.04 / 83.49 ± 48.72	5.45 ± 7.38 / 48.75 ± 38.82
	30	5.57 ± 2.30 / 52.53 ± 26.60	8.58 ± 2.08 / 71.15 ± 30.97	6.77 ± 7.78 / 56.02 ± 48.11
	35	7.42 ± 4.44 / 74.85 ± 81.66	8.64 ± 2.58 / 69.82 ± 48.35	6.72 ± 7.07 / 41.88 ± 40.02
	40	12.82 ± 11.61 / 77.48 ± 74.78	10.28 ± 3.92 / 77.36 ± 59.83	7.62 ± 6.46 / 50.37 ± 44.90
	45	10.41 ± 10.22 / 66.70 ± 74.35	10.36 ± 7.74 / 117.71 ± 68.31	8.23 ± 7.52 / 64.02 ± 53.38
	50	12.07 ± 11.50 / 63.09 ± 59.14	15.67 ± 16.77 / 121.45 ± 99.81	8.84 ± 7.68 / 52.39 ± 38.96
	55	14.42 ± 11.36 / 83.63 ± 71.66	19.35 ± 20.34 / 159.67 ± 102.47	8.77 ± 7.87 / 52.26 ± 38.94
	60	13.35 ± 10.86 / 76.20 ± 75.12	25.24 ± 21.26 / 133.62 ± 88.23	9.01 ± 6.16 / 48.00 ± 28.75
	65	13.88 ± 9.99 / 62.70 ± 50.67	30.42 ± 20.92 / 80.18 ± 57.58	10.79 ± 5.12 / 48.08 ± 19.66
	70	15.05 ± 9.19 / 57.95 ± 43.45	33.49 ± 23.23 / 64.33 ± 48.38	13.63 ± 6.51 / 47.88 ± 22.79
	75	16.00 ± 9.84 / 67.77 ± 39.64	31.03 ± 24.61 / 68.03 ± 48.92	11.99 ± 6.38 / 46.63 ± 24.32
Triple	5	3.47 ± 4.85 / 195.65 ± 168.27	18.16 ± 14.47 / 574.99 ± 648.93	2.23 ± 2.42 / 207.51 ± 176.42
	10	2.88 ± 4.13 / 105.30 ± 92.55	11.34 ± 8.61 / 269.00 ± 267.07	2.17 ± 2.33 / 97.50 ± 74.79
	15	2.45 ± 4.52 / 63.50 ± 49.90	7.28 ± 4.49 / 156.22 ± 138.15	2.06 ± 2.39 / 60.06 ± 41.15
	20	3.28 ± 3.94 / 53.45 ± 41.84	8.01 ± 4.69 / 97.37 ± 117.14	2.36 ± 1.38 / 53.48 ± 31.96
	25	3.22 ± 4.43 / 95.09 ± 121.11	7.35 ± 7.07 / 90.29 ± 78.36	1.76 ± 1.22 / 48.79 ± 23.61
	30	6.77 ± 6.36 / 49.42 ± 42.54	10.21 ± 8.25 / 83.65 ± 85.00	4.53 ± 2.52 / 51.04 ± 22.71
	35	6.75 ± 5.24 / 40.53 ± 34.47	10.29 ± 8.09 / 73.28 ± 58.42	5.86 ± 2.62 / 47.92 ± 29.81
	40	6.39 ± 4.75 / 42.23 ± 37.11	9.22 ± 4.18 / 69.25 ± 52.72	8.72 ± 7.92 / 49.97 ± 48.33
	45	7.03 ± 6.50 / 69.62 ± 84.56	14.44 ± 18.02 / 152.10 ± 139.55	10.00 ± 10.12 / 60.67 ± 63.93
	50	8.52 ± 7.00 / 54.01 ± 50.62	15.26 ± 18.84 / 126.06 ± 95.27	10.60 ± 9.77 / 57.67 ± 48.85
	55	6.91 ± 4.07 / 47.67 ± 45.70	21.84 ± 20.87 / 166.52 ± 108.29	14.66 ± 17.05 / 73.49 ± 79.32
	60	8.91 ± 6.84 / 50.17 ± 44.82	22.06 ± 19.40 / 145.94 ± 104.14	12.82 ± 14.59 / 56.18 ± 61.84
	65	10.23 ± 5.97 / 46.38 ± 27.64	22.28 ± 19.09 / 102.05 ± 71.15	11.77 ± 11.72 / 44.57 ± 40.08
	70	12.75 ± 6.85 / 41.78 ± 23.03	24.58 ± 18.08 / 85.96 ± 57.13	13.79 ± 9.19 / 44.93 ± 32.03
	75	11.37 ± 7.12 / 41.49 ± 22.25	28.40 ± 19.15 / 88.45 ± 55.52	12.56 ± 10.33 / 44.39 ± 33.09

D.2 BREAKDOWNS OF TABLE 2

In this section, we provide the first-convergence results for different post-shift thresholds $\hat{b}$ in the Bullet Safety Gym tasks. We report P_{50} and P_{80} separately for each threshold in the single-threshold setting.

Table 11: Detailed first-convergence iterations for the Bullet Safety Gym tasks at each post-shift threshold $\hat{b}$ in the single-threshold setting. Each cell reports P_{50}/P_{80} over 20 runs. A percentile that does not converge within the evaluation horizon is reported at the maximum iteration.

Environment	$\hat{b}$	ZAPD	CCPO	CAPS
Ball-Circle	5	20 / 23	30 / 80	13.5 / 19.2
	15	10.5 / 13	25.5 / 31.2	12 / 14
	25	6 / 7	22 / 25.8	12 / 15
	35	1.5 / 2	20 / 22.4	12 / 13.2
	45	1 / 2	20 / 23	12 / 13.2
	55	6 / 7	18 / 20	10 / 11
	65	9 / 9.2	16 / 17.2	9 / 9.2
	75	9 / 10	14 / 15.2	8.5 / 9
Car-Circle	5	18 / 26.2	29 / 39.2	31 / 44.4
	15	16 / 18.2	23.5 / 30	18 / 29.6
	25	12.5 / 19.2	22 / 27	14.5 / 22
	35	3 / 5.2	22.5 / 28	12 / 16
	45	5 / 6	23 / 25.2	11 / 14.2
	55	10 / 12.2	22 / 26	10 / 13
	65	13 / 16.2	23 / 29.4	8.5 / 13
	75	14.5 / 22.2	24 / 35	10.5 / 16.6
Drone-Circle	5	21.5 / 27.8	57.5 / 71.2	42 / 56.2
	15	19.5 / 30.8	57 / 61.4	40 / 50.2
	25	8.5 / 15	51 / 57.2	35.5 / 40
	35	2 / 4.2	50 / 58.4	36 / 39
	45	2 / 5	52.5 / 58.4	36 / 39.4
	55	7.5 / 10.4	54.5 / 63	33.5 / 42
	65	15 / 24	53 / 58.2	32 / 37.2
	75	29.5 / 36.2	49.5 / 54.2	32 / 38
Drone-Run	5	31 / 57	29.5 / 34.2	27 / 37
	15	33.5 / 110.4	27 / 40	26 / 42.8
	25	30.5 / 78.4	25.5 / 35.2	24 / 92.6
	35	26.5 / 47.4	51.5 / 64	33.5 / 48.4
	45	29 / 59.8	58.5 / 120	44 / 59
	55	25.5 / 58.6	47 / 60.2	25.5 / 35.6
	65	27 / 45.2	58.5 / 82.8	33 / 57.4
	75	29 / 47.8	68 / 92.4	33.5 / 49.8

D.3 BREAKDOWNS OF TABLE 3

In this section, we provide the immediate approximation errors for different post-shift thresholds $\hat{b}$ in CostLQR and Swimmer. We also present the breakdown between reward error and cost error for both linear and NN policies.

Table 12: Detailed immediate approximation error (%) for CostLQR and Swimmer, **Linear actor**. Each cell reports reward error (mean ± standard deviation) / cost error (mean ± standard deviation) over 20 runs.

			C-PG		PPO-L	
Environment	Setting	$\hat{b}$	ZAPD	CAPS	ZAPD	CAPS
CostLQR	Single	6	6.49 ± 7.33 / 3.54 ± 2.88	9.42 ± 3.41 / 22.76 ± 8.03	7.76 ± 5.22 / 6.40 ± 4.09	8.86 ± 5.34 / 28.42 ± 14.24
		6.375	4.94 ± 4.49 / 4.37 ± 3.28	5.72 ± 3.01 / 15.14 ± 7.53	6.19 ± 4.20 / 6.59 ± 4.00	5.89 ± 4.70 / 21.27 ± 12.67
		6.75	4.14 ± 3.51 / 5.00 ± 3.61	3.99 ± 2.91 / 9.28 ± 6.38	5.43 ± 4.10 / 6.85 ± 4.22	5.01 ± 3.90 / 15.50 ± 10.80
		7.125	3.71 ± 3.23 / 5.24 ± 3.92	3.43 ± 2.66 / 6.31 ± 4.03	5.05 ± 4.20 / 7.29 ± 4.50	5.02 ± 3.11 / 11.42 ± 8.82
		7.875	3.68 ± 2.85 / 5.87 ± 4.64	3.14 ± 2.85 / 6.49 ± 5.11	5.08 ± 4.16 / 9.30 ± 5.46	5.09 ± 2.74 / 8.54 ± 6.69
		8.25	3.73 ± 2.76 / 6.77 ± 5.36	3.22 ± 2.95 / 9.30 ± 5.92	5.08 ± 4.13 / 10.77 ± 6.66	5.14 ± 2.79 / 9.56 ± 6.31
		8.625	3.76 ± 2.70 / 8.08 ± 6.33	3.33 ± 3.06 / 13.12 ± 5.68	5.07 ± 4.11 / 13.12 ± 7.69	5.21 ± 2.86 / 11.00 ± 7.03
		9	3.77 ± 2.64 / 9.58 ± 7.29	3.50 ± 3.14 / 17.22 ± 5.41	5.05 ± 4.08 / 16.02 ± 9.04	5.26 ± 2.96 / 13.12 ± 7.59
	Double	6	6.08 ± 5.79 / 3.12 ± 2.62	5.22 ± 3.47 / 5.78 ± 3.33	5.48 ± 4.39 / 2.70 ± 1.61	5.77 ± 4.11 / 6.95 ± 5.04
		6.375	3.92 ± 3.66 / 3.38 ± 2.81	3.92 ± 3.66 / 3.38 ± 2.81	4.99 ± 2.77 / 4.15 ± 3.05	4.99 ± 2.77 / 4.15 ± 3.05
		6.75	3.84 ± 3.39 / 4.05 ± 3.35	4.79 ± 4.11 / 7.07 ± 3.77	5.18 ± 4.18 / 3.43 ± 2.13	5.10 ± 3.34 / 6.20 ± 3.68
		7.125	3.94 ± 3.05 / 4.92 ± 3.98	5.65 ± 4.58 / 11.53 ± 3.76	5.20 ± 4.31 / 4.37 ± 3.45	5.41 ± 4.05 / 10.19 ± 4.68
		7.875	3.64 ± 2.76 / 5.61 ± 3.80	7.05 ± 5.25 / 19.20 ± 3.43	5.01 ± 3.90 / 6.09 ± 4.31	6.10 ± 4.70 / 17.79 ± 4.29
		8.25	3.62 ± 2.70 / 5.46 ± 3.79	7.48 ± 5.49 / 22.42 ± 3.30	5.02 ± 3.88 / 6.25 ± 4.63	6.46 ± 4.99 / 21.04 ± 4.12
		8.625	2.89 ± 2.37 / 5.56 ± 3.28	2.89 ± 2.37 / 5.56 ± 3.28	4.96 ± 2.60 / 9.45 ± 6.42	4.96 ± 2.60 / 9.45 ± 6.42
		9	3.63 ± 2.65 / 5.87 ± 3.81	2.90 ± 2.36 / 8.28 ± 4.96	5.05 ± 3.90 / 7.18 ± 5.77	4.97 ± 2.59 / 9.19 ± 6.46
	Triple	6	6.08 ± 5.79 / 3.12 ± 2.62	5.22 ± 3.47 / 5.78 ± 3.33	5.48 ± 4.39 / 2.70 ± 1.61	5.77 ± 4.11 / 6.95 ± 5.04
		6.375	3.92 ± 3.66 / 3.38 ± 2.81	3.92 ± 3.66 / 3.38 ± 2.81	4.99 ± 2.77 / 4.15 ± 3.05	4.99 ± 2.77 / 4.15 ± 3.05
		6.75	3.84 ± 3.39 / 4.05 ± 3.35	4.79 ± 4.11 / 7.07 ± 3.77	5.18 ± 4.18 / 3.43 ± 2.13	5.10 ± 3.34 / 6.20 ± 3.68
		7.125	3.71 ± 3.23 / 5.24 ± 3.92	5.65 ± 4.58 / 11.53 ± 3.76	5.05 ± 4.20 / 7.29 ± 4.50	5.41 ± 4.05 / 10.19 ± 4.68
		7.875	3.68 ± 2.85 / 5.87 ± 4.64	3.14 ± 2.85 / 6.49 ± 5.11	5.08 ± 4.16 / 9.30 ± 5.46	5.09 ± 2.74 / 8.54 ± 6.69
		8.25	3.62 ± 2.70 / 5.46 ± 3.79	3.22 ± 2.95 / 9.30 ± 5.92	5.02 ± 3.88 / 6.25 ± 4.63	5.14 ± 2.79 / 9.56 ± 6.31
		8.625	2.89 ± 2.37 / 5.56 ± 3.28	2.89 ± 2.37 / 5.56 ± 3.28	4.96 ± 2.60 / 9.45 ± 6.42	4.96 ± 2.60 / 9.45 ± 6.42
		9	3.63 ± 2.65 / 5.87 ± 3.81	2.90 ± 2.36 / 8.28 ± 4.96	5.05 ± 3.90 / 7.18 ± 5.77	4.97 ± 2.59 / 9.19 ± 6.46
Swimmer	Single	1	29.60 ± 24.72 / 31.71 ± 20.26	12.55 ± 6.69 / 115.41 ± 59.75	6.37 ± 3.77 / 12.77 ± 10.31	17.77 ± 2.50 / 147.13 ± 19.61
		1.5	14.05 ± 14.30 / 26.55 ± 21.35	8.06 ± 5.39 / 66.91 ± 44.06	5.67 ± 2.92 / 9.12 ± 5.45	9.19 ± 2.31 / 65.52 ± 13.13
		2	6.51 ± 5.07 / 23.62 ± 18.85	5.75 ± 3.68 / 32.17 ± 29.48	2.59 ± 1.49 / 8.25 ± 4.66	4.52 ± 2.19 / 24.12 ± 9.85
		3	3.79 ± 2.41 / 19.42 ± 18.30	5.16 ± 3.13 / 23.95 ± 14.73	1.85 ± 2.36 / 7.02 ± 5.75	1.80 ± 1.39 / 17.29 ± 6.56
		3.5	4.48 ± 4.13 / 22.78 ± 20.76	5.22 ± 3.87 / 30.64 ± 15.49	14.12 ± 10.86 / 20.01 ± 19.29	2.80 ± 1.94 / 29.12 ± 5.62
		4	6.26 ± 8.83 / 28.88 ± 30.06	5.45 ± 4.56 / 38.50 ± 15.11	31.69 ± 11.32 / 70.42 ± 55.87	3.97 ± 2.04 / 37.91 ± 4.93
	Double	1	16.16 ± 12.86 / 28.67 ± 16.81	6.87 ± 4.67 / 33.02 ± 29.30	3.78 ± 2.68 / 8.79 ± 5.74	9.78 ± 2.12 / 48.28 ± 8.99
		1.5	4.64 ± 3.10 / 21.43 ± 11.64	4.64 ± 3.10 / 21.43 ± 11.64	2.12 ± 1.57 / 5.07 ± 3.11	2.12 ± 1.57 / 5.07 ± 3.11
		2	4.60 ± 3.27 / 20.74 ± 19.48	4.76 ± 3.67 / 25.27 ± 16.60	2.52 ± 1.95 / 9.86 ± 6.17	2.71 ± 1.68 / 25.53 ± 4.52
		3	4.81 ± 4.10 / 24.65 ± 17.91	7.66 ± 5.09 / 49.47 ± 12.55	2.31 ± 1.40 / 7.35 ± 5.03	7.67 ± 1.78 / 50.38 ± 3.01
		3.5	4.61 ± 3.00 / 23.03 ± 17.80	4.61 ± 3.00 / 23.03 ± 17.80	1.75 ± 1.32 / 6.66 ± 3.53	9.31 ± 1.75 / 57.48 ± 2.58
		4	3.45 ± 2.50 / 22.03 ± 17.85	4.42 ± 2.95 / 21.62 ± 14.87	1.58 ± 1.48 / 8.26 ± 6.91	1.41 ± 0.94 / 12.76 ± 6.72
	Triple	1	16.16 ± 12.86 / 28.67 ± 16.81	6.87 ± 4.67 / 33.02 ± 29.30	3.78 ± 2.68 / 8.79 ± 5.74	9.78 ± 2.12 / 48.28 ± 8.99
		1.5	4.64 ± 3.10 / 21.43 ± 11.64	4.64 ± 3.10 / 21.43 ± 11.64	2.12 ± 1.57 / 5.07 ± 3.11	2.12 ± 1.57 / 5.07 ± 3.11
		2	6.51 ± 5.07 / 23.62 ± 18.85	4.76 ± 3.67 / 25.27 ± 16.60	2.59 ± 1.49 / 8.25 ± 4.66	2.71 ± 1.68 / 25.53 ± 4.52
		3	4.81 ± 4.10 / 24.65 ± 17.91	5.16 ± 3.13 / 23.95 ± 14.73	2.31 ± 1.40 / 7.35 ± 5.03	1.80 ± 1.39 / 17.29 ± 6.56
		3.5	4.61 ± 3.00 / 23.03 ± 17.80	4.61 ± 3.00 / 23.03 ± 17.80	1.75 ± 1.32 / 6.66 ± 3.53	2.80 ± 1.94 / 29.12 ± 5.62
		4	3.45 ± 2.50 / 22.03 ± 17.85	4.42 ± 2.95 / 21.62 ± 14.87	1.58 ± 1.48 / 8.26 ± 6.91	1.41 ± 0.94 / 12.76 ± 6.72

Table 13: Detailed immediate approximation error (%) for CostLQR and Swimmer, **NN actor**. Each cell reports reward error (mean ± standard deviation) / cost error (mean ± standard deviation) over 20 runs.

			C-PG		PPO-L	
Environment	Setting	$\hat{b}$	ZAPD	CAPS	ZAPD	CAPS
CostLQR	Single	6	17.25 ± 22.38 / 7.88 ± 5.68	8.47 ± 5.73 / 23.81 ± 14.37	23.32 ± 21.48 / 10.07 ± 8.14	7.65 ± 4.70 / 28.41 ± 15.64
		6.375	12.43 ± 14.66 / 8.00 ± 5.58	6.29 ± 4.37 / 16.78 ± 12.52	12.37 ± 11.13 / 7.76 ± 7.78	4.96 ± 3.96 / 20.85 ± 14.72
		6.75	9.51 ± 10.37 / 8.62 ± 5.71	6.40 ± 4.09 / 12.56 ± 9.32	6.14 ± 4.89 / 7.48 ± 7.94	4.48 ± 3.36 / 14.50 ± 13.53
		7.125	7.71 ± 7.42 / 9.09 ± 6.22	6.76 ± 4.29 / 10.15 ± 7.09	3.78 ± 2.86 / 7.77 ± 8.56	4.58 ± 3.24 / 10.08 ± 11.70
		7.875	5.94 ± 4.48 / 10.53 ± 6.92	7.38 ± 4.90 / 10.44 ± 6.97	3.04 ± 2.56 / 9.00 ± 10.78	5.15 ± 3.22 / 8.91 ± 7.94
		8.25	5.68 ± 3.74 / 11.66 ± 7.41	7.63 ± 5.24 / 12.22 ± 8.08	3.84 ± 2.90 / 9.98 ± 12.56	5.32 ± 3.35 / 10.72 ± 6.82
		8.625	5.52 ± 3.39 / 13.08 ± 8.95	7.81 ± 5.49 / 14.22 ± 8.92	6.83 ± 5.90 / 11.26 ± 14.93	5.50 ± 3.45 / 13.19 ± 5.80
		9	5.52 ± 3.28 / 14.96 ± 11.20	8.00 ± 5.68 / 16.60 ± 9.46	11.53 ± 10.50 / 13.38 ± 17.85	5.67 ± 3.56 / 15.54 ± 5.35
	Double	6	9.56 ± 9.77 / 5.32 ± 3.12	5.76 ± 4.55 / 6.72 ± 4.98	5.41 ± 4.78 / 3.92 ± 3.82	4.73 ± 3.92 / 7.56 ± 5.42
		6.375	7.13 ± 4.82 / 4.95 ± 3.84	7.13 ± 4.82 / 4.95 ± 3.84	4.91 ± 2.65 / 3.78 ± 3.89	4.91 ± 2.65 / 3.78 ± 3.89
		6.75	5.98 ± 4.84 / 6.04 ± 3.95	8.86 ± 5.75 / 7.78 ± 4.92	3.94 ± 3.05 / 4.65 ± 4.60	5.72 ± 3.37 / 5.83 ± 3.49
		7.125	5.59 ± 3.99 / 6.61 ± 4.85	9.96 ± 6.40 / 12.17 ± 5.39	9.71 ± 8.66 / 5.51 ± 5.76	6.34 ± 4.26 / 10.06 ± 3.65
		7.875	7.08 ± 6.73 / 10.74 ± 8.10	11.46 ± 7.08 / 20.37 ± 5.01	4.11 ± 2.93 / 9.70 ± 7.07	7.32 ± 5.16 / 17.87 ± 4.39
		8.25	6.22 ± 5.54 / 11.22 ± 8.20	12.06 ± 7.29 / 23.73 ± 4.80	3.60 ± 2.58 / 10.22 ± 7.71	7.69 ± 5.34 / 21.29 ± 4.20
		8.625	6.29 ± 4.01 / 11.38 ± 8.37	6.29 ± 4.01 / 11.38 ± 8.37	5.28 ± 3.06 / 9.26 ± 8.46	5.28 ± 3.06 / 9.26 ± 8.46
		9	5.63 ± 3.76 / 12.33 ± 8.99	6.40 ± 4.20 / 12.00 ± 8.39	3.37 ± 2.40 / 11.95 ± 9.26	5.41 ± 3.11 / 10.06 ± 7.21
	Triple	6	9.56 ± 9.77 / 5.32 ± 3.12	5.76 ± 4.55 / 6.72 ± 4.98	5.41 ± 4.78 / 3.92 ± 3.82	4.73 ± 3.92 / 7.56 ± 5.42
		6.375	7.13 ± 4.82 / 4.95 ± 3.84	7.13 ± 4.82 / 4.95 ± 3.84	4.91 ± 2.65 / 3.78 ± 3.89	4.91 ± 2.65 / 3.78 ± 3.89
		6.75	5.98 ± 4.84 / 6.04 ± 3.95	8.86 ± 5.75 / 7.78 ± 4.92	3.94 ± 3.05 / 4.65 ± 4.60	5.72 ± 3.37 / 5.83 ± 3.49
		7.125	7.71 ± 7.42 / 9.09 ± 6.22	9.96 ± 6.40 / 12.17 ± 5.39	3.78 ± 2.86 / 7.77 ± 8.56	6.34 ± 4.26 / 10.06 ± 3.65
		7.875	5.94 ± 4.48 / 10.53 ± 6.92	7.38 ± 4.90 / 10.44 ± 6.97	3.04 ± 2.56 / 9.00 ± 10.78	5.15 ± 3.22 / 8.91 ± 7.94
		8.25	6.22 ± 5.54 / 11.22 ± 8.20	7.63 ± 5.24 / 12.22 ± 8.08	3.60 ± 2.58 / 10.22 ± 7.71	5.32 ± 3.35 / 10.72 ± 6.82
		8.625	6.29 ± 4.01 / 11.38 ± 8.37	6.29 ± 4.01 / 11.38 ± 8.37	5.28 ± 3.06 / 9.26 ± 8.46	5.28 ± 3.06 / 9.26 ± 8.46
		9	5.63 ± 3.76 / 12.33 ± 8.99	6.40 ± 4.20 / 12.00 ± 8.39	3.37 ± 2.40 / 11.95 ± 9.26	5.41 ± 3.11 / 10.06 ± 7.21
Swimmer	Single	1	26.58 ± 30.80 / 58.26 ± 52.67	11.20 ± 6.36 / 117.06 ± 98.24	15.03 ± 8.88 / 19.14 ± 13.48	16.21 ± 2.19 / 151.67 ± 20.30
		1.5	19.81 ± 27.34 / 51.67 ± 43.76	9.24 ± 5.45 / 72.26 ± 64.95	6.74 ± 5.24 / 9.61 ± 6.18	8.49 ± 2.05 / 67.38 ± 13.50
		2	10.64 ± 10.71 / 43.50 ± 35.58	7.97 ± 4.23 / 47.49 ± 38.74	2.12 ± 1.46 / 7.41 ± 4.98	3.95 ± 1.91 / 26.07 ± 10.17
		3	6.93 ± 4.22 / 31.69 ± 22.90	6.97 ± 3.66 / 37.90 ± 24.72	1.41 ± 1.02 / 9.52 ± 6.90	2.04 ± 1.58 / 15.94 ± 6.78
		3.5	7.00 ± 4.60 / 27.10 ± 18.26	6.49 ± 3.65 / 42.13 ± 24.53	2.37 ± 1.47 / 17.76 ± 11.29	3.84 ± 1.81 / 27.99 ± 5.81
		4	7.60 ± 4.66 / 22.70 ± 18.77	6.26 ± 3.74 / 48.10 ± 22.47	3.65 ± 1.46 / 28.15 ± 17.48	5.58 ± 1.78 / 36.98 ± 5.08
	Double	1	14.88 ± 19.62 / 34.96 ± 33.50	9.37 ± 4.92 / 49.61 ± 41.30	6.94 ± 4.83 / 7.50 ± 5.05	8.19 ± 2.25 / 52.48 ± 11.18
		1.5	8.06 ± 3.79 / 30.80 ± 26.19	8.06 ± 3.79 / 30.80 ± 26.19	1.91 ± 1.28 / 6.71 ± 3.15	1.91 ± 1.28 / 6.71 ± 3.15
		2	7.13 ± 4.18 / 24.70 ± 21.78	6.95 ± 3.66 / 31.37 ± 23.51	1.71 ± 1.64 / 15.58 ± 8.38	3.25 ± 2.01 / 23.62 ± 5.60
		3	8.03 ± 6.25 / 49.74 ± 43.72	6.10 ± 4.36 / 51.85 ± 19.17	1.44 ± 1.17 / 4.90 ± 4.57	8.57 ± 1.90 / 49.07 ± 3.73
		3.5	6.67 ± 4.20 / 44.11 ± 36.78	5.75 ± 4.90 / 59.38 ± 16.58	1.37 ± 0.75 / 4.80 ± 3.27	1.37 ± 0.75 / 4.80 ± 3.27
		4	6.82 ± 4.01 / 39.10 ± 31.64	5.66 ± 5.18 / 64.81 ± 14.36	1.25 ± 0.99 / 6.24 ± 4.15	1.50 ± 1.24 / 12.58 ± 5.17
	Triple	1	14.88 ± 19.62 / 34.96 ± 33.50	9.37 ± 4.92 / 49.61 ± 41.30	6.94 ± 4.83 / 7.50 ± 5.05	8.19 ± 2.25 / 52.48 ± 11.18
		1.5	8.06 ± 3.79 / 30.80 ± 26.19	8.06 ± 3.79 / 30.80 ± 26.19	1.91 ± 1.28 / 6.71 ± 3.15	1.91 ± 1.28 / 6.71 ± 3.15
		2	10.64 ± 10.71 / 43.50 ± 35.58	6.95 ± 3.66 / 31.37 ± 23.51	2.12 ± 1.46 / 7.41 ± 4.98	3.25 ± 2.01 / 23.62 ± 5.60
		3	8.03 ± 6.25 / 49.74 ± 43.72	6.10 ± 4.36 / 51.85 ± 19.17	1.44 ± 1.17 / 4.90 ± 4.57	2.04 ± 1.58 / 15.94 ± 6.78
		3.5	6.67 ± 4.20 / 44.11 ± 36.78	6.49 ± 3.65 / 42.13 ± 24.53	1.37 ± 0.75 / 4.80 ± 3.27	1.37 ± 0.75 / 4.80 ± 3.27
		4	6.82 ± 4.01 / 39.10 ± 31.64	6.26 ± 3.74 / 48.10 ± 22.47	1.25 ± 0.99 / 6.24 ± 4.15	1.50 ± 1.24 / 12.58 ± 5.17

D.4 BREAKDOWNS OF TABLE 4

In this section, we provide the first-convergence results for different post-shift thresholds $\hat{b}$ in CostLQR and Swimmer. We report P_{50} and P_{80} separately for each threshold, base method, and policy parameterization in the single-threshold setting.

Table 14: Detailed first-convergence iterations for CostLQR and Swimmer at each post-shift threshold $\hat{b}$ in the single-threshold setting. Each cell reports P_{50} / P_{80} over 20 runs. A percentile that does not converge within the evaluation horizon is reported as the maximum iteration.

			C-PG			PPO-L		
Environment		$\hat{b}$	ZAPD (full)	ZAPD (primal)	CAPS	ZAPD (full)	ZAPD (primal)	CAPS
CostLQR	Linear	6	4 / 26	3 / 18	148 / 181	5 / 13	6 / 58	43 / 55
		6.375	3 / 12	3 / 24	94 / 149	3 / 9	3 / 41	31 / 40
		6.75	2 / 6	2 / 4	56 / 120	3 / 8	3 / 30	26 / 36
		7.125	2 / 12	2 / 4	2 / 30	3 / 13	4 / 28	13 / 24
		7.875	3 / 54	2 / 63	2 / 94	2 / 13	3 / 23	2 / 21
		8.25	3 / 54	3 / 56	12 / 118	2 / 12	3 / 21	4 / 18
		8.625	5 / 56	6 / 67	106 / 246	4 / 18	4 / 21	10 / 35
		9	11 / 69	7 / 67	208 / 311	16 / 24	17 / 24	16 / 35
	NN	6	10 / 38	10 / 82	101 / 151	11 / 23	30 / 39	34 / 41
		6.375	3 / 24	8 / 36	35 / 77	5 / 22	5 / 26	24 / 32
		6.75	10 / 33	8 / 39	13 / 69	3 / 11	3 / 17	19 / 23
		7.125	20 / 40	22 / 50	9 / 62	3 / 7	2 / 7	3 / 17
		7.875	17 / 38	24 / 70	18 / 67	2 / 12	2 / 15	3 / 17
		8.25	14 / 33	14 / 57	27 / 99	2 / 6	3 / 13	10 / 22
		8.625	10 / 33	13 / 63	36 / 109	4 / 15	5 / 13	15 / 24
		9	14 / 34	12 / 50	83 / 117	8 / 17	7 / 15	16 / 26
Swimmer	Linear	1	56 / 246	64 / 435	393 / 649	157 / 230	272 / 312	281 / 311
		1.5	36 / 116	37 / 386	265 / 402	6 / 20	6 / 96	118 / 137
		2	36 / 100	52 / 179	157 / 272	3 / 4	3 / 3	48 / 73
		3	3 / 71	3 / 128	41 / 245	3 / 3	3 / 29	33 / 51
		3.5	59 / 242	29 / 128	134 / 355	4 / 11	4 / 6	44 / 54
		4	67 / 252	34 / 208	221 / 392	7 / 46	6 / 52	48 / 51
	NN	1	169 / 1000	259 / 370	273 / 478	35 / 163	248 / 299	229 / 263
		1.5	156 / 1000	237 / 541	164 / 379	10 / 18	13 / 119	97 / 117
		2	280 / 1000	232 / 839	171 / 288	3 / 8	3 / 45	63 / 72
		3	225 / 680	511 / 1000	312 / 1000	2 / 3	2 / 3	27 / 52
		3.5	207 / 544	442 / 976	618 / 1000	13 / 62	7 / 10	58 / 67
		4	307 / 1000	594 / 1000	818 / 1000	37 / 66	16 / 76	65 / 75

E MULTIPLE-CONSTRAINT EXPERIMENTS

The experiments in Section 5 focus on the single-constraint setting ($m = 1$) to keep the main text evaluation focused on broad comparisons across environments and methods. Since the formulation and zero-shot updates in Sections 3 and 4 are developed for general m, we additionally evaluate a two-constraint setting here to verify their applicability when multiple constraint thresholds shift simultaneously.

E.1 EXPERIMENTAL SETUP

We use PPO-L on CostLQR with two constraints. The first cost is the control cost used in the single-constraint experiments,

$$c_{1,t} = a_t^\top R a_t, \tag{55}$$

and the second penalizes variation between consecutive actions,

$$c_{2,t} = \|a_t - a_{t-1}\|_2^2. \tag{56}$$

At the first time step, we set $a_{-1} = \mathbf{0}$ and define $c_{2,0} = 0$; the above cost is applied from $t = 1$ onward. Accordingly, the CMDP has constraint returns $\boldsymbol{\mathcal{J}}(\theta) = [\mathcal{J}_1(\theta), \mathcal{J}_2(\theta)]^\top$ and threshold vector $\boldsymbol{b} = [b_1, b_2]^\top$.

We consider the two threshold settings

$$\boldsymbol{b}^{(1)} = \begin{bmatrix} 6 \\ 16 \end{bmatrix}, \qquad \boldsymbol{b}^{(2)} = \begin{bmatrix} 7.5 \\ 17 \end{bmatrix}. \tag{57}$$

The relaxation experiment shifts $\boldsymbol{b}^{(1)} \to \boldsymbol{b}^{(2)}$, while the tightening experiment uses the reverse shift $\boldsymbol{b}^{(2)} \to \boldsymbol{b}^{(1)}$. The remaining CostLQR and PPO-L implementation settings are the same as those described in Appendix C.4.

E.2 RESULTS

The post-shift reference values for $\mathcal{R}$, $\mathcal{J}_1$, and $\mathcal{J}_2$ are obtained from five independent runs trained from scratch under the corresponding post-shift threshold vector, using the same procedure as in Appendix C.3. We use the first-convergence criterion of Appendix C.3, requiring the reward and both constraint returns to lie within 5% of their post-shift reference values. For this two-constraint experiment, we use 25 runs, formed by pairing five independent pre-shift runs with five post-shift runs each. The post-shift evaluation horizon is 400 iterations.

ZAPD attains the lowest P_{50} and P_{80} in both directions. These results show that ZAPD extends beyond the single-constraint experiments of Section 5 and remains effective when multiple constraint thresholds shift jointly.

Table 15: First-convergence iterations for two-constraint CostLQR with PPO-L. Each cell reports P_{50}/P_{80}; lower is better. Best values are bold and shaded.

Shift	ZAPD	No ZAPD
Relax	**36 / 92**	215 / 255
Tighten	**14 / 29**	106 / 151