LOCU: Löwdin-Orthogonalized Constraint Updates for Multi-Constraint Policy Optimization

Joonyoung Lim
Department of Information Convergence Engineering
Pusan National University
ingyuh1015@pusan.ac.kr

Younghwan Yoo*
School of Computer Science and Engineering
Pusan National University
ymomo@pusan.ac.kr

Abstract

We investigate multi-constraint policy optimization in constrained Markov decision processes (CMDPs), where interactions among constraint gradients often yield ill-conditioned update directions, causing numerically fragile steps and unintended cancellation or redundant overlap in constraint corrections. To address this, we propose LOCU, which mitigates numerical instability arising from constraint interactions by applying symmetric Löwdin orthogonalization to the natural gradients of the reward and constraints under the Fisher–Rao metric. Combined with Fisher-geometric screening and near-collinear compression, LOCU yields a compact reduced space in which the trust region becomes a Euclidean ball and constraints are handled symmetrically through a low-dimensional active-set solve. For infeasible iterates, LOCU constructs a Pareto-descent direction which preserves the local feasibility of satisfied constraints within the same reduced formulation. Experiments show that LOCU remains stable under high constraint coupling, narrow feasible regions, and infeasible starts, and generalizes across different cost critic architectures. Code is available at: `GitHub Repository`.

1 Introduction

Reinforcement learning has achieved notable success in sequential decision making, but deployment in safety-critical domains requires policies to additionally satisfy explicit operational constraints rather than maximize reward alone [20, 21, 1, 2]. Constrained Markov decision processes (CMDPs) formalize this requirement by optimizing return subject to cost limits. In practical scenarios, agents could face several constraints simultaneously, such as velocity, collision, or resource limit in autonomous driving or robotic control. Existing safe policy optimization methods are commonly proposed and evaluated in single-constraint regimes, and their behavior can degrade when multiple constraints interact.

The core difficulty with multiple constraints is not merely computational overhead, but the *interaction geometry* among constraints [18]. Existing approaches expose one or more of the following failures (i) unstable updates under near-dependent constraint gradients or severe scale disparity, (ii) weakened effective updates under constraint interaction, caused by cancellation, dominance by a single direction, or projection-order dependence, and (iii) inefficient feasibility recovery after constraint violations,

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

as noted in prior work [14]. The problem worsens under many or near-parallel constraints, making stability depend on hyperparameter tuning rather than geometry-consistent structure.

A geometric perspective clarifies why these issues arise and how to address them. Policy optimization can be analyzed on a Riemannian manifold whose metric arises as the second-order local approximation of the KL divergence, known as the Fisher–Rao metric [4, 3]. Reward and constraint objectives define tangent directions in natural-gradient coordinates [12], and each linearized constraint induces a halfspace normal to its constraint gradient. With multiple constraints, feasible updates lie in the intersection of these halfspaces and the local Fisher-metric ellipsoid [1, 28]. Redundant or ill-conditioned constraint directions under this metric make both first- and second-order updates fragile and conservative, an effect amplified by finite-sample noise or critic error.

Building on this geometric insight, we introduce the Löwdin Orthogonalized Constraint Update (LOCU), a second-order framework for multi-constraint RL that mitigates constraint coupling in Fisher-consistent coordinates. LOCU constructs a Fisher-orthonormal basis from the natural gradients of the reward and constraints via symmetric Löwdin orthogonalization [17]. This transformation maps the Fisher-metric ellipsoid into a spherical trust region in a reduced Euclidean space, making constraint interactions explicit and numerically stable. For large constraint sets, LOCU employs Fisher-geometric screening to prune non-binding or redundant constraints. Furthermore, it handles infeasibility with a geometry-aligned recovery step that seeks to reduce violations while preserving satisfied constraints in the linearized local model. The update operates purely at the policy optimization layer, ensuring compatibility with various critic architectures.

We conduct experiments with multi-constrained tasks from [8, 15] applying the constraint configurations introduced by [29]. In evaluations under narrow feasible regions and strong constraint coupling, LOCU satisfies all constraints while limiting return drops to around 21% in both settings, compared to 26–49% and 13–32% drops with frequent violations in baselines, performing consistently from infeasible starts and generalizing across different cost critic architectures. These results demonstrate the effectiveness of the proposed method in maintaining stable policy optimization by symmetrically handling constraint interactions under the Fisher–Rao geometry.

2 Related Work

Existing constrained RL methods differ in how they incorporate multiple constraints into policy optimization and maintain feasibility during actor updates. These design choices become tightly coupled with the representation of constraint interactions. When constraint interactions are not handled in a geometry-consistent manner under the Fisher–Rao metric, stable learning often relies heavily on environment-specific tuning of numerous algorithmic choices and hyperparameters.

Primal-dual Methods. Lagrangian-based methods adopt a primal-dual formulation where constraints are managed via Lagrange multipliers co-optimized with policy parameters [23]. However, with multiple constraints, these methods aggregate costs into a scalar, obscuring the directional structure of individual constraint gradients. Depending on alignment, constraint gradients may cancel and suppress effective updates, or reinforce and destabilize multiplier dynamics, causing oscillations and training instability. As the number of constraints increases, performance becomes increasingly sensitive to multiplier initialization and tuning can be "as hard as solving the RL problem itself" [19].

Recent work has attempted to address gradient-level interference in multi-constraint settings by analyzing directional interactions and applying shaping rules to improve update consistency [29]. While this mitigates some interference issues within the Lagrangian framework, its reliance on Euclidean angle-based similarity may not faithfully reflect the true correlations under the Fisher–Rao geometry [12]. Furthermore, it does not alleviate the burden of tuning an expanding set of hyperparameters as constraints grow.

A separate line of work handles dual variables by solving a quadratic program (QP) or projection defined by linearized constraints within a KL-based trust region [1]. In multi-constraint regimes, feasibility recovery is either not explicitly addressed once iterates leave the local feasible region [1], or handled via repeated constraint-wise projections that shrink the effective reward step and introduce imbalances as the number of constraints grows [28]. Moreover, the active set is unknown a priori, these updates rely on generic QP solvers. Under linear dependence or near-collinearity among constraint gradients, the resulting QP can be ill-conditioned, leading to solver failures or

high-variance solutions where it is difficult to distinguish intrinsic constraint degeneracy from solver-induced numerical errors.

Recent work in risk-constraint RL proposed gradient integration mechanisms to manage infeasibility start and recovery in multi-constraint settings and further extend these ideas toward global convergence guarantees [14, 15]. Despite these advances, general applicability across constrained RL is largely limited to gradient integration, while the absence of a geometry-consistent formulation and continued dependence on generic solvers hinder scalability and stability as the number of constraints and interactions grows.

Primal Methods. Primal methods can be naturally extended to multi-constraint settings by switching between reward optimization and constraint minimization [26]. However, as the number of constraints increases, handling one constraint at a time leads to slower convergence and inconsistent updates. Furthermore, substantial correlations between adaptive weights and stochastic gradients are known to bias the update step and cause convergence failure [31]. To address such instabilities, recent work proposes a soft-switching strategy [9], which reduces oscillations and accelerates convergence by removing conflicting components between reward and constraint gradients. While it improves stability in single-constraint scenarios, effectiveness in multi-constraint settings remains unverified.

For a geometric discussion of multi-constraint update pathologies in representative primal–dual and projection-based methods, see Appendix C.1.

3 Preliminaries

3.1 Constrained Markov Decision Processes

A CMDP is defined by $(\mathcal{S}, \mathcal{A}, P, r, \{c_k\}_{k=1}^K, \rho, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, $P(\cdot \mid s, a)$ is the transition probability, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $c_k : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_{\geq 0}$ are K cost functions that quantify constraint violations [2]. The initial state is $s_0 \sim \rho$ and the discount factor is $\gamma \in (0, 1)$. A stochastic policy $\pi(a \mid s)$ induces a trajectory distribution $\mathbb{P}_\pi$ over $(s_0, a_0, s_1, \dots)$. Given a policy π, its discounted state–action distribution is

$$d^\pi(s, a) := (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \, \mathbb{P}_\pi(s_t = s, a_t = a). \tag{1}$$

Using this distribution, define the discounted return $J_R(\pi) := \mathbb{E}_{s_0 \sim \rho}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$, and for each cost k define $J_{C_k}(\pi) := \mathbb{E}_{s_0 \sim \rho}[\sum_{t=0}^{\infty} \gamma^t c_k(s_t, a_t)]$. The CMDP objective is

$$\max_\pi \; J_R(\pi) \quad \text{s.t.} \quad J_{C_k}(\pi) \leq b_k, \forall k. \tag{2}$$

Assumption 3.1 (Slater condition)**.** There exists a policy $\bar{\pi}$ in the considered policy class such that

$$J_{C_k}(\bar{\pi}) < b_k, \qquad \forall k \in \{1, \dots, K\}. \tag{3}$$

This feasibility condition is standard in constrained policy optimization [1].

3.2 Policy Gradients for Reward and Constraints

Consider a differentiable policy π_θ parameterized by $\theta \in \mathbb{R}^n$. Let $A_R^\pi(s, a) := Q_R^\pi(s, a) - V_R^\pi(s)$. The policy gradient for the reward objective is

$$\nabla_\theta J_R(\pi_\theta) = \tfrac{1}{1-\gamma} \, \mathbb{E}_{(s,a) \sim d^{\pi_\theta}} \left[\nabla_\theta \log \pi_\theta(a \mid s) \, A_R^{\pi_\theta}(s, a) \right]. \tag{4}$$

Similarly, each cost objective admits a policy gradient obtained by replacing r with c_k. We denote $g_r := \nabla_\theta J_R(\pi_\theta)$ and $g_{c_k} := \nabla_\theta J_{C_k}(\pi_\theta)$.

3.3 Trust-Region Geometry and Natural Gradient

Trust-region updates are measured by the average KL divergence under the discounted state visitation distribution,

$$\bar{D}_{\mathrm{KL}}(\pi_\theta, \pi_{\theta'}) := \mathbb{E}_{s \sim d^{\pi_\theta}} \left[\mathrm{KL}(\pi_\theta(\cdot \mid s) \,\|\, \pi_{\theta'}(\cdot \mid s)) \right]. \tag{5}$$

For a small displacement $\Delta\theta := \theta' - \theta$, a second-order Taylor expansion gives

$$\bar{D}_{\mathrm{KL}}(\pi_\theta, \pi_{\theta+\Delta\theta}) = \frac{1}{2}\Delta\theta^\top F\,\Delta\theta + o(\|\Delta\theta\|^2), \tag{6}$$

where F is the Fisher information matrix (FIM),

$$F := \mathbb{E}_{(s,a)\sim d^{\pi_\theta}}\left[\nabla_\theta \log \pi_\theta(a \mid s)\, \nabla_\theta \log \pi_\theta(a \mid s)^\top\right]. \tag{7}$$

Equation (6) induces the Fisher–Rao geometry, where the FIM F defines the inner product $\langle u, v\rangle_F := u^\top F v$ and the corresponding norm $\|u\|_F := \sqrt{u^\top F u}$. Since the FIM F is always positive semi-definite, we assume it to be positive definite ($F \succ 0$) in what follows so that F^{-1} and the Fisher inner product are well-defined (in practice, this can be enforced by diagonal damping).

Under this geometry, the natural gradient is defined as the steepest-ascent direction with respect to the Fisher–Rao metric:

$$\tilde{g} := F^{-1} g. \tag{8}$$

We denote natural gradients for the reward and costs as $\tilde{g}_r := F^{-1} g_r$ and $\tilde{g}_{c_k} := F^{-1} g_{c_k}$.

3.4 Local Multi-Constraint Trust-Region Subproblem

Under a KL-based trust region, Eq. (2) is locally approximated by first-order expansions of J_R and J_{C_k} at θ together with the Fisher quadratic approximation (6). Define the current constraint slacks as $\Delta_k := b_k - J_{C_k}(\pi_\theta)$. The local trust-region CMDP step is

$$\begin{aligned} \max_{\Delta\theta\in\mathbb{R}^n} \quad & g_r^\top \Delta\theta \\ \text{s.t.} \quad & g_{c_k}^\top \Delta\theta \le \Delta_k, \quad k = 1, \dots, K, \\ & \tfrac{1}{2}\,\|\Delta\theta\|_F^2 \le \delta. \end{aligned} \tag{9}$$

Problem (9) is a convex program. When the current iterate is strictly feasible, i.e., $\Delta_k > 0$ for all k and $\delta > 0$, the point $\Delta\theta = 0$ lies in the interior of the feasible set. In this case, strong duality holds and the optimum is characterized by the Karush–Kuhn–Tucker (KKT) conditions and the associated dual variables [6].

Lemma 3.2. *Problem* (9) *admits an optimal solution* $\Delta\theta^\star$ *such that* $\Delta\theta^\star \in \mathrm{span}\{\tilde{g}_r, \tilde{g}_{c_1}, \dots, \tilde{g}_{c_K}\}$.

Proof. See Appendix A.2.1. □

3.5 Symmetric Löwdin orthogonalization

When working with multiple update directions, it is often useful to replace a correlated basis by an orthonormal one in a way that is symmetric and insensitive to the ordering of directions. A classical construction is the symmetric orthogonalization of Löwdin [17].

Let $S = [s_1, \dots, s_p] \in \mathbb{R}^{n\times p}$ be a matrix whose columns are linearly independent directions. Define the Gram matrix $G := S^\top S \in \mathbb{R}^{p\times p}$. Since $\mathrm{rank}(S) = p$, we have $G \succ 0$ and the principal inverse square root $G^{-1/2}$ is well-defined. The symmetric Löwdin orthogonalization constructs

$$E := S\,G^{-1/2} \in \mathbb{R}^{n\times p}, \qquad E^\top E = G^{-1/2} S^\top S\, G^{-1/2} = I. \tag{10}$$

so the columns of E are orthonormal and $\mathrm{span}(E) = \mathrm{span}(S)$.

Unlike Gram–Schmidt, the Löwdin construction is symmetric: it uses the symmetric inverse square root of $G = S^\top S$ and does not privilege any particular column order. This order-invariant structure is the property used by LOCU when constructing reduced coordinates from reward and constraint directions.

4 Proposed Method

We focus on the local multi-constraint trust-region subproblem in (9). By Lemma 3.2, it suffices to search over the subspace spanned by the reward and constraint natural-gradient directions. We construct Fisher-orthonormal reduced coordinates for this subspace and solve an equivalent low-dimensional convex program. Practical implementation details are provided in Appendix A.4.

4.1 Fisher-orthonormal reduced coordinates via Löwdin orthogonalization

Motivated by the symmetric Löwdin orthogonalization introduced in Section 3.5, we apply the same construction to the reward and constraint natural-gradient directions under the Fisher–Rao metric.

Let $\tilde{g}_r := F^{-1}g_r$ and $\tilde{g}_{c_k} := F^{-1}g_{c_k}$ denote the reward and cost natural gradients. Fix any set of m representative constraint indices $\{i_1, \dots, i_m\} \subseteq \{1, \dots, K\}$, and form the direction matrix

$$S := \begin{bmatrix} \tilde{g}_r & \tilde{g}_{c_{i_1}} & \cdots & \tilde{g}_{c_{i_m}} \end{bmatrix} \in \mathbb{R}^{n\times(m+1)}. \tag{11}$$

Define the Gram matrix under the Fisher inner product, which we call the *Fisher Gram matrix* for convenience:

$$O := S^\top F S \in \mathbb{R}^{(m+1)\times(m+1)}. \tag{12}$$

We construct the symmetric Löwdin basis

$$E := SO^{-1/2} \in \mathbb{R}^{n\times(m+1)}. \tag{13}$$

Lemma 4.1. *If $O \succ 0$ (or if O is regularized to be positive definite), then*

$$E^\top F E = I. \tag{14}$$

Proof is in Appendix A.2.2.

Lemma 4.1 implies that the columns of E form an orthonormal basis under the Fisher inner product. We restrict the update to $\mathrm{span}(S)$ and write $\Delta\theta = Ew$ with $w \in \mathbb{R}^{m+1}$. Let $R := \sqrt{2\delta}$. Since $E^\top F E = I$ by (14), the quadratic KL trust region reduces to

$$\tfrac{1}{2}\Delta\theta^\top F \Delta\theta = \tfrac{1}{2} w^\top w = \tfrac{1}{2}\|w\|_2^2, \tag{15}$$

and hence becomes the Euclidean ball $\|w\|_2 \le R$ in the reduced coordinates.

Define the reduced constraint normals and the reduced-space objective coefficients in the Löwdin basis by

$$u := E^\top g_r, \qquad C_k^\top := E^\top g_{c_k} \in \mathbb{R}^{m+1}. \tag{16}$$

Here, we use the symbol u as the objective coefficient vector of the reduced program. In the feasible phase, $u = E^\top g_r$ as above; in the feasibility-recovery phase (Section 4.4), we simply replace u by the recovery direction constructed directly in the same reduced space.

Let $\mathcal{K} \subseteq \{1, \dots, K\}$ be the set of constraints enforced in the current update, and collect the corresponding rows into $C \in \mathbb{R}^{|\mathcal{K}|\times(m+1)}$ and $\Delta \in \mathbb{R}^{|\mathcal{K}|}$ with $\Delta_k = b_k - J_{C_k}(\pi_\theta)$. Then the reduced program is

$$\max_{w\in\mathbb{R}^{m+1}} \; u^\top w \quad \text{s.t.} \quad \|w\|_2 \le R, \;\; Cw \le \Delta. \tag{17}$$

Proposition 4.2. *Let e_j be the jth canonical basis vector in $\mathbb{R}^{m+1}$, and let the columns of S be ordered as in* (11)*. Then*

$$E^\top g_r = O^{1/2} e_1, \quad E^\top g_{c_{i_j}} = O^{1/2} e_{j+1}, \quad j = 1, \dots, m. \tag{18}$$

Proof is in Appendix A.2.3.

Theorem 4.3. *Assume $O \succ 0$ (or O is regularized so that $O \succ 0$). Fix any enforced constraint index set $\mathcal{K}$ and any representative set $\{i_1, \dots, i_m\}$ used to form S, O, E as in* (11)–(13)*. Then the change of variables $\Delta\theta = Ew$ gives a one-to-one correspondence between $\mathrm{span}(S)$ and $\mathbb{R}^{m+1}$ and transforms the trust-region CMDP subproblem* (9) *restricted to $\Delta\theta \in \mathrm{span}(S)$ and constraints $k \in \mathcal{K}$ into the reduced program* (17)*. In particular, any optimizer $w^\star$ of* (17) *yields a valid update $\Delta\theta^\star = Ew^\star$ in the original parameter space.*

Proof is in Appendix A.2.4.

Unlike Gram–Schmidt orthogonalization, which propagates and accumulates perturbations through a sequential process, Löwdin orthogonalization in Proposition 4.2 distributes perturbations symmetrically across the eigenspace. We analyze these perturbation structures in Appendix C.2.1.

4.2 Constraint screening and basis compression

However, directly working with all constraint directions can be expensive at scale and can be numerically unstable when constraint directions are strongly aligned in Fisher–Rao geometry. Therefore, LOCU screens locally non-binding/degenerate constraints and compresses near-collinear directions for basis construction, while still enforcing every retained constraint in the reduced optimization.

With Eqs. (11) and (12), we start from full direction with reward and K constraints $S_{\text{full}} \in \mathbb{R}^{n\times(K+1)}$ and its Fisher Gram matrix $O_{\text{full}} \in \mathbb{R}^{(K+1)\times(K+1)}$. For each constraint k, define the Fisher norm and normalized slack

$$\nu_k := \sqrt{(O_{\text{full}})_{k+1,k+1}}, \qquad \mu_k := \Delta_k/\nu_k, \tag{19}$$

and for each pair (i, j) define the Fisher cosine similarity among constraint natural gradients

$$\varrho_{ij} := \frac{(O_{\text{full}})_{i+1,j+1}}{\nu_i\nu_j}. \tag{20}$$

Equivalently, with Fisher-unit directions $u_k := \tilde{g}_{c_k}/\nu_k$ (so that $\|u_k\|_F = 1$), we have $\varrho_{ij} = \langle u_i, u_j\rangle_F$ and the kth linearized constraint is $\langle u_k, \Delta\theta\rangle_F \le \mu_k$.

4.2.1 Constraint screening

We perform worst-case Fisher screening and exclude non-binding and near-degenerate constraints.

Lemma 4.4. *For any $\Delta\theta$ satisfying $\|\Delta\theta\|_F \le R$,*

$$g_{c_k}^\top \Delta\theta \le \nu_k R. \tag{21}$$

Proof is in Appendix A.2.5.

Proposition 4.5. *If $\Delta_k \ge \nu_k R$, then constraint k cannot be active inside the trust region $\frac{1}{2}\Delta\theta^\top F\Delta\theta \le \delta$ and can be excluded from $\mathcal{K}$ without changing feasibility within that trust region.*

Proof is in Appendix A.2.6.

In addition, constraints with $\nu_k \le \varepsilon_{\text{nz}}$ are excluded from $\mathcal{K}$ to avoid unstable normalization in (19), since $\tilde{g}_{c_k}$ is nearly degenerate in Fisher–Rao geometry.

4.2.2 Basis clustering

We compress near-collinear constraint directions only to stabilize basis construction; all constraints in $\mathcal{K}$ remain enforced in the reduced program.

We first build an undirected graph on $\mathcal{K}$ by connecting each pair (i, j) that satisfies $|\varrho_{ij}| \ge 1 - \varepsilon_{\text{dup}}$. Let $\{\mathcal{G}_1, \dots, \mathcal{G}_m\}$ be the connected components of this graph. Then, for each group $\mathcal{G}_\ell$, we select a representative index by $i_\ell \in \arg\min_{k\in\mathcal{G}_\ell} \mu_k$. If all μ_k in the group are nonnegative, this rule selects the tightest constraint in Fisher-normalized units. If some μ_k are negative, it selects the most violated constraint in the same units. The reduced basis matrix S in (11) is then formed using $\{i_1, \dots, i_m\}$.

Lemma 4.6. *Let u_i, u_j be Fisher-unit vectors with $\langle u_i, u_j\rangle_F = \rho \in [-1, 1]$. Then for any $\Delta\theta$ with $\|\Delta\theta\|_F \le R$,*

$$\min_{s\in\{+1,-1\}} \left|\langle u_i, \Delta\theta\rangle_F - s\,\langle u_j, \Delta\theta\rangle_F\right| \le \sqrt{2(1-|\rho|)}\,R. \tag{22}$$

In particular, if $|\rho| \ge 1 - \varepsilon_{\text{dup}}$, the discrepancy is at most $\sqrt{2\varepsilon_{\text{dup}}}\,R$. One may take $s = +1$ when $\rho \ge 0$ and $s = -1$ when $\rho < 0$.

Proof is in Appendix A.2.7.

Lemma 4.6 shows that near-collinear directions produce similar linearized constraint changes within the trust region, up to a sign, motivating approximate basis compression using representative directions. Importantly, all constraints in $\mathcal{K}$ remain explicitly enforced in (17) via the rows of C.

4.3 Solving the reduced program

We solve (17) in the reduced coordinates. In the feasible case (all slacks in Δ are nonnegative), we set $u = E^\top g_r$. If the halfspaces are inactive, the maximizer over the ball has the closed form

$$w_0 = R\frac{u}{\|u\|_2}, \tag{23}$$

whenever $\|u\|_2 \neq 0$. If $Cw_0 \leq \Delta$, we accept $w^\star = w_0$. Otherwise, we use KKT conditions. There exist $\lambda \in \mathbb{R}^{|\mathcal{K}|}$ and $\beta \in \mathbb{R}$ such that

$$-u + C^\top \lambda + 2\beta w = 0, \qquad \lambda \geq 0, \qquad \beta \geq 0, \tag{24}$$

together with primal feasibility, complementary slackness, and $\|w\|_2 \leq R$. For an active set $\mathcal{J} \subseteq \{1, \ldots, |\mathcal{K}|\}$ with $\beta > 0$, stationarity yields

$$w = \frac{1}{2\beta}\big(u - C_{\mathcal{J}}^\top \lambda_{\mathcal{J}}\big), \tag{25}$$

and enforcing $C_{\mathcal{J}} w = \Delta_{\mathcal{J}}$ gives the reduced linear system

$$C_{\mathcal{J}} C_{\mathcal{J}}^\top \lambda_{\mathcal{J}} = C_{\mathcal{J}} u - 2\beta \Delta_{\mathcal{J}}. \tag{26}$$

The linear system in (26) involves only the multipliers associated with the working set $\mathcal{J}$ and therefore has dimension $|\mathcal{J}|$. Since $|\mathcal{J}| \leq |\mathcal{K}|$, its size is bounded by the number of retained constraints, independently of the policy parameter dimension n. The active-set routine and the one-dimensional search for β are given in Appendix A.3.

4.4 Feasibility recovery via Pareto descent in the orthonormal space

When some constraints are violated ($\Delta_k < 0$ for at least one $k \in \mathcal{K}$), we use a recovery objective that prioritizes feasibility improvement in the linearized model, while preserving the linearized feasibility of already satisfied constraints.

Let $\mathcal{V} := \{i \in \{1, \ldots, |\mathcal{K}|\} \mid \Delta_i < 0\}$ and $\mathcal{H} := \{i \in \{1, \ldots, |\mathcal{K}|\} \mid \Delta_i \geq 0\}$, with indices defined according to the row ordering of C induced by $\mathcal{K}$. We compute the simplex weights by solving the following problem:

$$\min_{\alpha \in \mathbb{R}^{|\mathcal{V}|}} \ \|C_{\mathcal{V}}^\top \alpha\|_2^2 \quad \text{s.t.} \quad \alpha \geq 0,\ \mathbf{1}^\top \alpha = 1. \tag{27}$$

and define the Pareto descent direction

$$\bar{g}_p := -C_{\mathcal{V}}^\top \alpha. \tag{28}$$

Lemma 4.7. *If $\bar{g}_p \neq 0$ defined by* (27)–(28)*, then for every $i \in \mathcal{V}$,*

$$C_i^\top \bar{g}_p < 0. \tag{29}$$

Proof is in Appendix A.2.8.

We set the recovery objective to $u = \bar{g}_p$ and solve

$$\max_{w \in \mathbb{R}^{m+1}} \ \bar{g}_p^\top w \quad \text{s.t.} \quad \|w\|_2 \leq R, \quad C_{\mathcal{H}} w \leq \Delta_{\mathcal{H}}. \tag{30}$$

The point $w = 0$ is feasible since $\Delta_{\mathcal{H}} \geq 0$, so (30) preserves the linearized feasibility of the already satisfied constraints.

Computational complexity. The proposed method starts by computing natural gradients for the reward and constraints. Each gradient computation requires one backpropagation over the entire batch. Computing a natural gradient with conjugate gradient (CG) further requires T_{cg} iterations per gradient, each consisting of one or two backpropagations depending on the implementation. Therefore, the computational cost can be written as $\mathcal{O}\big((K+1) \cdot T_{\text{cg}} \cdot \texttt{BackProp}\big)$.

After obtaining the natural gradients, the Fisher Gram matrix is constructed from K^2 pairwise inner products between them, resulting in a cost of $\mathcal{O}(K^2 \cdot \texttt{InnerProd})$. Screening and clustering are

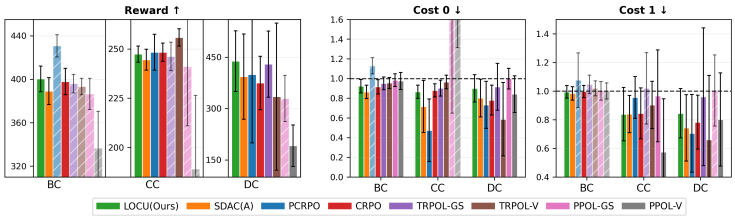


Figure 1: Final evaluation on the v2 Circle tasks, with high-velocity and boundary constraints.

performed at a cost of $\mathcal{O}(K + |\mathcal{K}|^2)$, followed by Löwdin orthogonalization of the m representative directions at a cost of $\mathcal{O}(m^2 \cdot \texttt{InnerProd} + m^3)$. Solving the reduced program with the active-set method requires $\mathcal{O}(T_{\text{active}} \cdot |\mathcal{K}|^3)$ operations, where T_{active} is the number of active-set iterations, while an additional simplex QP over the violated constraints is solved in the recovery case. Since $m \leq |\mathcal{K}| \leq K$, the computational cost of LOCU procedure is bounded by $\mathcal{O}(K^2 \cdot \texttt{InnerProd} + T_{\text{active}} \cdot K^3)$.

Both back-propagation and inner product scale linearly with the size of policy parameters n. However, back-propagation involves forward and backward passes over the entire batch, whereas inner product requires only element-wise multiplication and summation. Consequently, the overall computational cost is dominated by natural-gradient computation, which is also required by other second-order multi-constraint methods, leaving LOCU with only marginal additional overhead. Further discussion related to computational cost is provided in Appendix C.3.1.

5 Evaluation and Results

5.1 Tasks and Baselines

Tasks We evaluated on two settings. First, we use Circle tasks from Bullet-Safety-Gym [8] with Ball, Car, and Drone agents, denoted as BC, CC, and DC, under the multi-constraint setting of [29]. The task is to follow a circular trajectory, where costs are incurred for leaving the boundary region or violating upper and lower velocity limits. We vary constraint thresholds and extend the tasks with redundancy variants [29] to control feasible-set geometry and induce near-collinear constraint directions.

Second, we evaluate legged locomotion tasks introduced by [15] on the bipedal and quadrupedal robots Cassie (Agility Robotics) and Laikago (Unitree), respectively. The objective is to track a commanded velocity while enforcing three constraints on body balance, height, and foot contact timing. To further test heterogeneous constraint handling, we add a goal-navigation variant that retains the three locomotion-stability constraints and introduces three qualitatively different constraints on hazard contact, speed, and energy, which was originally on the reward side. This variant requires the agent to reach a goal while avoiding hazards and satisfying operational limits. Both versions are initialized from infeasible states with simultaneous constraint violations under high-dimensional observations, making these tasks suitable for assessing feasibility recovery and compatibility with alternative critic types. For locomotion tasks, all methods use the same quantile-based cost critic.

Baselines The baselines include the actor update of SDAC [14], denoted as SDAC(A), which uses a second-order trust-region QP update with gradient integration for recovery, while all other components are aligned with our setup. We also evaluate two primal methods, PCRPO [9] and CRPO [26], alongside PPO and TRPO Lagrangian with PID-style multiplier updates [23], each with and without gradient shaping (GS) [29], denoted as PPOL-GS, PPOL-V, TRPOL-GS, and TRPOL-V, respectively. For the final task, we further apply Clarabel, a modern interior-point conic solver [7], to both LOCU's reduced QP and the full-dimensional QP. Detailed descriptions for both task and baselines are provided in Appendix B.

5.2 Results

Results are averaged over five training seeds and 100 evaluation episodes with 10 rollout seeds. For the Circle tasks, we additionally reduce evaluation bias by averaging over the five checkpoints at the end of each training run. All costs are normalized by their corresponding limits; values above 1

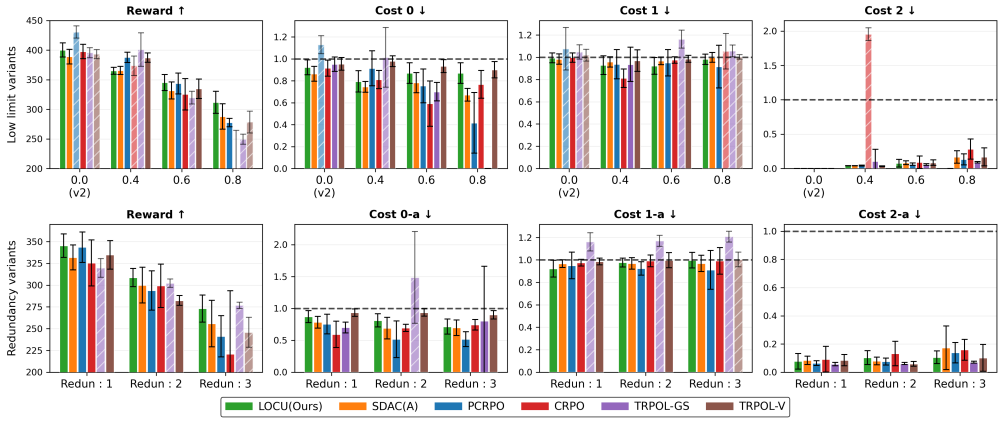


Figure 2: Final evaluation results on BC-v3 variants. Top: the low-velocity requirement $\kappa_{\mathrm{lo}} \in \{0.0, 0.4, 0.6, 0.8\}$. Bottom: redundancy variants with increasing numbers of same-type boundary and velocity constraints. The suffix "–a" marks the tightest constraint among the same type.

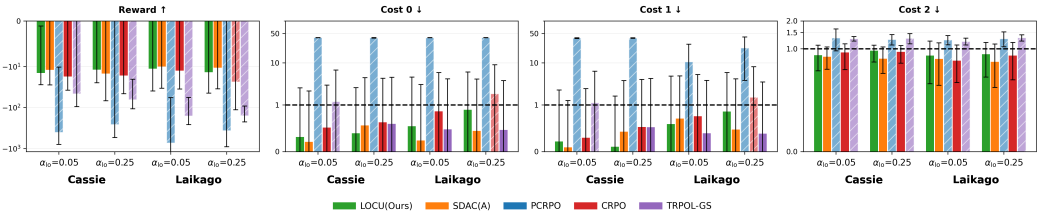


Figure 3: Final evaluation on the locomotion tasks Cassie and Laikago with three simultaneous constraints. Costs 0, 1, and 2 denote the balance, height, and foot timing constraints. Mean–std constraints use $\alpha_{\mathrm{lo}} \in \{0.05, 0.25\}$. Cost axes are shown on a log scale.

indicate constraint violations. In the bar plots, violating cost bars and the corresponding reward bar are rendered with reduced opacity and hatching. Error bars indicate standard deviation. Additional results of ablation studies and full learning curves are provided in Appendix C.4 and C.5.

Fig. 1 summarizes v2 results on BC, CC, and DC. LOCU consistently satisfies all constraint limits across agents while achieving the highest returns on BC and DC among methods satisfying all constraints, and remaining competitive on CC. The PPOL variants exhibit instability in this multi-constraint setting, where the trained policy fails to reliably track the circular objective. Therefore, we omit PPOL-GS and -V from subsequent experiments.

Fig. 2 summarizes the BC-v3 results under feasibility-tightening and redundancy variants. Feasibility tightening includes low-velocity constraints, which shrink the feasible set and reshape early learning by forcing the agent to increase speed before stable circle tracking, thereby delaying reward improvement. This effect is stronger when training starts from infeasible policies: trust-region and primal methods prioritize feasibility recovery for longer as thresholds tighten, while Lagrangian methods may preserve higher return through multiplier adaptation but show less stable constraint satisfaction when aggressively growing multipliers destabilize optimization. The redundancy variants build on the base v3 setting with a single low-velocity coefficient of 0.6, simultaneously shrinking the feasible set and inducing near-collinear constraint directions. Across both tightening and redundancy variants, LOCU maintains stable constraint satisfaction while limiting return degradation relative to the baselines.

Figure 3 summarizes the results on Cassie and Laikago with three simultaneous constraints. LOCU and SDAC(A) achieve the best returns while consistently satisfying all constraint limits, whereas PCRPO does not reliably converge in our multi-constraint locomotion setup. Since PCRPO reduces oscillations through soft-switching, we interpret this result as indicating that additional stabilization may be required in coupled multi-constraint regimes with intermittent violations and cross-constraint scale mismatch.

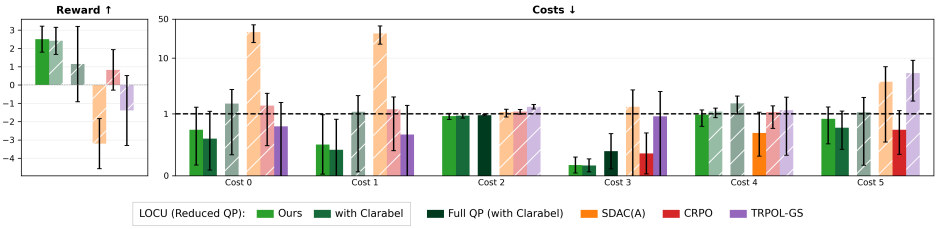


Figure 4: Final evaluation on the Cassie goal task with six constraints. Costs 0, 1, and 2 denote the balance, height, and foot timing constraints. Costs 3, 4, and 5 denote the speed, hazard contact, and energy constraints, respectively. Cost axes are shown on a log scale and 0.25 is used for α_{lo}.

Additional results on the goal-navigation task, including two variants using Clarabel, are summarized in Figure 4. Here, SDAC(A) misallocates recovery weights as the QP becomes severely ill-conditioned. Unlike the original locomotion task, where the diagonal terms are comparable, body tilt and height constraints violate their limits by roughly 30 times, resulting in much larger diagonal terms than weak-gradient constraints such as energy or obstacle contact. The dual update aims to reduce all active constraints without considering the violation levels, leading to large recovery weights with little correction and negligible weights on the constraints that require correction most. Full QP variants which use the same set of natural gradients exhibit similar behavior, although solving with Clarabel partially improves recovery. Notably, the solver flagged only 2 of approximately 24,900 updates across five seeds as inaccurate, indicating that the remaining degradation stems from severe ill-conditioning of the QP itself rather than from frequent solver failures. LOCU avoids this by screening out inactive or degenerate constraints, and constructing an orthogonal basis of the retained gradients. Despite some constraint violations, the Reduced QP variants which use LOCU's orthogonal basis perform closest to LOCU in both return and constraint satisfaction. This suggests that LOCU's orthogonal basis construction can be integrated into existing algorithms to enhance numerical stability, while its proposed active-set solver can be replaced with alternative solvers depending on computational and real-time requirements. We provide the additional robustness test on goal-navigation task with redundancy in Appendix C.4.3.

6 Limitations and future work

Like other second-order trust-region methods [1], LOCU provides local feasibility control but no global convergence guarantee. A possible approach is to reinterpret KL-induced Fisher–Rao geometry through a Bregman-proximal mirror-descent perspective over the convex occupancy-measure CMDP, which may provide a route to primal-dual convergence analysis. Also, computing natural-gradient directions and the Fisher Gram matrix for all constraints may raise computational-efficiency concerns. However, as discussed in Appendix C.3.1, parallelized constraint processing can mitigate this issue by better utilizing GPU throughput. It may also be useful to extend natural gradient calculation with curvature approximations to further reduce cost of CG iteration and Fisher-related computations.

7 Conclusion

We introduced LOCU, a second-order framework for multi-constraint policy optimization that mitigates constraint coupling in a geometrically consistent manner. By applying symmetric Löwdin orthogonalization to natural gradients under the Fisher–Rao metric, LOCU yields an orthonormal reduced coordinate system in which the KL trust region becomes a Euclidean ball and constraint interactions are handled in a numerically stable way. Across multi-constraint benchmarks with tight feasible regions, redundant constraints, and infeasible starts, LOCU achieves stable constraint handling with competitive returns across initial feasibility conditions and critic architectures.

Acknowledgments and Disclosure of Funding

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) through the Information Technology Research Center program (IITP-2026-RS-2023-00260098) and the Artificial Intelligence Convergence Innovation Human Resources Development program (IITP-2026-RS-2023-00254177).

We thank Google TPU Research Cloud (TRC) for granting us access to TPUs. Additionally, J.L. thanks Nixon Son for the discussion that inspired the initial idea behind this work.

References

[1] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pages 22–31. PMLR, 2017.

[2] Eitan Altman. *Constrained Markov decision processes*. Routledge, 2021.

[3] Shun-ichi Amari. *Information geometry and its applications*, volume 194. Springer, 2016.

[4] Nihat Ay, Jürgen Jost, Hông Vân Lê, and Lorenz Schwachhöfer. *Information geometry*, volume 64. Springer, 2017.

[5] Lukas Biewald. Experiment tracking with weights and biases, 2020. URL `https://www.wandb.com/`. Software available from wandb.com.

[6] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[7] Paul J Goulart and Yuwen Chen. Clarabel: An interior-point solver for conic programs with quadratic objectives (2024). *arXiv preprint arXiv:2405.12762*.

[8] Sven Gronauer. Bullet-safety-gym: A framework for constrained reinforcement learning. Technical report, mediaTUM, 2022.

[9] Shangding Gu, Bilgehan Sel, Yuhao Ding, Lu Wang, Qingwei Lin, Ming Jin, and Alois Knoll. Balance reward and safety optimization for safe reinforcement learning: A perspective of gradient manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 21099–21106, 2024.

[10] 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 *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL `https://openreview.net/forum?id=WZmlxIuIGR`.

[11] 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.

[12] Sham M Kakade. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.

[13] Danial Kamran, Thiago D. Simão, Qisong Yang, Canmanie T. Ponnambalam, Johannes Fischer, Matthijs T. J. Spaan, and Martin Lauer. A modern perspective on safe automated driving for different traffic dynamics using constrained reinforcement learning. In *ITSC*, pages 4017–4023. IEEE, 2022. doi: 10.1109/ITSC55140.2022.9921907.

[14] Dohyeong Kim, Kyungjae Lee, and Songhwai Oh. Trust region-based safe distributional reinforcement learning for multiple constraints. *Advances in neural information processing systems*, 36:19908–19939, 2023.

[15] Dohyeong Kim, Taehyun Cho, Seungyub Han, Hojun Chung, Kyungjae Lee, and Songhwai Oh. Spectral-risk safe reinforcement learning with convergence guarantees. *Advances in Neural Information Processing Systems*, 37:92465–92498, 2024.

[16] Zuxin Liu, Zijian Guo, Haohong Lin, Yihang Yao, Jiacheng Zhu, Zhepeng Cen, Hanjiang Hu, Wenhao Yu, Tingnan Zhang, Jie Tan, and Ding Zhao. Datasets and benchmarks for offline safe reinforcement learning. *Journal of Data-centric Machine Learning Research*, 2024.

[17] Per-Olov Löwdin. On the non-orthogonality problem connected with the use of atomic wave functions in the theory of molecules and crystals. *The Journal of Chemical Physics*, 18(3):365–375, 1950.

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

[19] Santiago Paternain, Luiz Chamon, Miguel Calvo-Fullana, and Alejandro Ribeiro. Constrained reinforcement learning has zero duality gap. *Advances in Neural Information Processing Systems*, 32, 2019.

[20] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

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

[22] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4):637–672, 2020. doi: 10.1007/s12532-020-00179-2. URL `https://doi.org/10.1007/s12532-020-00179-2`.

[23] Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by pid lagrangian methods. In *International Conference on Machine Learning*, pages 9133–9143. PMLR, 2020.

[24] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

[25] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

[26] Tengyu Xu, Yingbin Liang, and Guanghui Lan. Crpo: A new approach for safe reinforcement learning with convergence guarantee. In *International Conference on Machine Learning*, pages 11480–11491. PMLR, 2021.

[27] Long Yang, Jiaming Ji, Juntao Dai, Linrui Zhang, Binbin Zhou, Pengfei Li, Yaodong Yang, and Gang Pan. Constrained update projection approach to safe policy optimization. *Advances in Neural Information Processing Systems*, 35:9111–9124, 2022.

[28] Tsung-Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J Ramadge. Projection-based constrained policy optimization. *arXiv preprint arXiv:2010.03152*, 2020.

[29] Yihang Yao, Zuxin Liu, Zhepeng Cen, Peide Huang, Tingnan Zhang, Wenhao Yu, and Ding Zhao. Gradient shaping for multi-constraint safe reinforcement learning. In *6th Annual Learning for Dynamics & Control Conference*, pages 25–39. PMLR, 2024.

[30] Yiming Zhang, Quan Vuong, and Keith Ross. First order constrained optimization in policy space. *Advances in Neural Information Processing Systems*, 33:15338–15349, 2020.

[31] Shiji Zhou, Wenpeng Zhang, Jiyan Jiang, Wenliang Zhong, Jinjie Gu, and Wenwu Zhu. On the convergence of stochastic multi-objective gradient manipulation and beyond. *Advances in Neural Information Processing Systems*, 35:38103–38115, 2022.

A Supplementary proofs and implementation details

This appendix provides (i) proofs omitted in the main text and (ii) additional derivations and implementation notes. To keep the section hierarchy consistent, *all omitted proofs* are grouped under Section A.2. The remaining sections provide supporting material (reduced-space KKT details, screening notes, recovery mechanics, empirical KL control, and projection-pathology analysis).

A.1 Notation and shapes

Let $\mathcal{K}$ be the enforced constraint index set after screening, and write $\bar{K} := |\mathcal{K}|$. Let $\{\mathcal{G}_1, \dots, \mathcal{G}_m\}$ be the near-collinear clusters and $\{i_1, \dots, i_m\}$ their representatives. Define

$$S \in \mathbb{R}^{n\times(m+1)}, \quad O := S^\top F S \in \mathbb{R}^{(m+1)\times(m+1)}, \quad E := SO^{-1/2} \in \mathbb{R}^{n\times(m+1)}.$$

Throughout the appendix we work with the generic reduced program

$$\max_{w\in\mathbb{R}^{m+1}} \; u^\top w \quad \text{s.t.} \quad \|w\|_2 \le R, \quad Cw \le \Delta,$$

where $u \in \mathbb{R}^{m+1}$ is the objective coefficient vector in reduced coordinates. In LOCU, u is set depending on the update mode:

$$u := \begin{cases} E^\top g_r & \text{(feasible case)}, \\ \bar{g}_p & \text{(infeasible case; Section 4.4)}. \end{cases}$$

For each enforced constraint index $k \in \mathcal{K}$, define the reduced normal

$$C_k^\top := E^\top g_{c_k} \in \mathbb{R}^{m+1}.$$

After fixing an ordering of $\mathcal{K}$, we assemble

$$C \in \mathbb{R}^{\bar{K}\times(m+1)}, \qquad \Delta \in \mathbb{R}^{\bar{K}}, \qquad R = \sqrt{2\delta}.$$

We use the Fisher inner product $\langle x, y\rangle_F := x^\top F y$ and norm $\|x\|_F := \sqrt{x^\top F x}$.

A.2 Proofs omitted in the main text

A.2.1 Lemma 3.2

Lemma 3.2. *Problem* (9) *admits an optimal solution* $\Delta\theta^\star$ *such that* $\Delta\theta^\star \in \operatorname{span}\{\tilde{g}_r, \tilde{g}_{c_1}, \dots, \tilde{g}_{c_K}\}$.

Proof. Let

$$\mathcal{U} := \operatorname{span}\{\tilde{g}_r, \tilde{g}_{c_1}, \dots, \tilde{g}_{c_K}\}.$$

Take any $\Delta\theta \in \mathbb{R}^n$ and decompose it as

$$\Delta\theta = \Delta\theta_\parallel + \Delta\theta_\perp, \qquad \Delta\theta_\parallel \in \mathcal{U}, \qquad \Delta\theta_\perp \in \mathcal{U}^{\perp_F},$$

where $\mathcal{U}^{\perp_F} := \{v : \langle u, v\rangle_F = 0,\ \forall u \in \mathcal{U}\}$.

First, the linearized reward objective depends only on $\Delta\theta_\parallel$:

$$g_r^\top \Delta\theta = (F\tilde{g}_r)^\top \Delta\theta = \tilde{g}_r^\top F \Delta\theta = \langle \tilde{g}_r, \Delta\theta_\parallel + \Delta\theta_\perp\rangle_F = \langle \tilde{g}_r, \Delta\theta_\parallel\rangle_F,$$

because $\tilde{g}_r \in \mathcal{U}$ and $\Delta\theta_\perp \in \mathcal{U}^{\perp_F}$.

Similarly, each constraint depends only on $\Delta\theta_\parallel$: for any k,

$$g_{c_k}^\top \Delta\theta = (F\tilde{g}_{c_k})^\top \Delta\theta = \tilde{g}_{c_k}^\top F \Delta\theta = \langle \tilde{g}_{c_k}, \Delta\theta_\parallel + \Delta\theta_\perp\rangle_F = \langle \tilde{g}_{c_k}, \Delta\theta_\parallel\rangle_F,$$

since $\tilde{g}_{c_k} \in \mathcal{U}$.

Finally, the trust-region quadratic term satisfies the Pythagorean decomposition:

$$\|\Delta\theta\|_F^2 = \langle \Delta\theta_\parallel + \Delta\theta_\perp, \Delta\theta_\parallel + \Delta\theta_\perp\rangle_F = \|\Delta\theta_\parallel\|_F^2 + \|\Delta\theta_\perp\|_F^2 \ge \|\Delta\theta_\parallel\|_F^2.$$

Therefore, if $\Delta\theta$ is feasible for (9), then $\Delta\theta_\parallel$ is also feasible (it does not increase the KL quadratic term and keeps all linear terms unchanged) and achieves the same objective value. Hence an optimal solution exists in $\mathcal{U}$. □

A.2.2 Lemma 4.1

Lemma 4.1. *If $O \succ 0$ (or if O is regularized to be positive definite), then*

$$E^\top F E = I.$$

Proof. Recall the definitions $E = SO^{-1/2}$ and $O = S^\top F S$. Then

$$E^\top F E = \big(SO^{-1/2}\big)^\top F \big(SO^{-1/2}\big) = O^{-1/2}\, S^\top F S\, O^{-1/2} = O^{-1/2}\, O\, O^{-1/2} = I.$$

□

A.2.3 Proposition 4.2

Proposition 4.2. *Let e_j be the jth canonical basis vector in $\mathbb{R}^{m+1}$, and let the columns of S be ordered as in* (11)*. Then*

$$E^\top g_r = O^{1/2} e_1, \qquad E^\top g_{c_{i_j}} = O^{1/2} e_{j+1}, \quad j = 1, \dots, m.$$

Proof. Let e_1 be the first canonical vector in $\mathbb{R}^{m+1}$. By construction of S in (11), we have $Se_1 = \tilde{g}_r$ and $g_r = F\tilde{g}_r$. Compute

$$S^\top g_r = S^\top F \tilde{g}_r = S^\top F (S e_1) = (S^\top F S) e_1 = O e_1.$$

Therefore,

$$E^\top g_r = (SO^{-1/2})^\top g_r = O^{-1/2} S^\top g_r = O^{-1/2}(O e_1) = O^{1/2} e_1.$$

For the representative constraint i_j, the $(j+1)$th column of S equals $\tilde{g}_{c_{i_j}}$, i.e., $Se_{j+1} = \tilde{g}_{c_{i_j}}$, and $g_{c_{i_j}} = F\tilde{g}_{c_{i_j}}$. Repeating the same steps gives

$$E^\top g_{c_{i_j}} = O^{1/2} e_{j+1}.$$

□

A.2.4 Theorem 4.3

Theorem 4.3. *Assume $O \succ 0$ (or O is regularized so that $O \succ 0$). Fix any enforced constraint index set $\mathcal{K}$ and any representative set $\{i_1, \dots, i_m\}$ used to form S, O, E as in* (11)–(13)*. Then the change of variables $\Delta\theta = Ew$ gives a one-to-one correspondence between* $\mathrm{span}(S)$ *and $\mathbb{R}^{m+1}$ and transforms the trust-region CMDP subproblem* (9) *restricted to $\Delta\theta \in \mathrm{span}(S)$ and constraints $k \in \mathcal{K}$ into the reduced program* (17)*. In particular, any optimizer $w^\star$ of* (17) *yields a valid update $\Delta\theta^\star = Ew^\star$ in the original parameter space.*

Proof. Since $O \succ 0$, the matrix $O^{-1/2}$ exists and is invertible. Hence $E = SO^{-1/2}$ has full column rank, and the map $w \mapsto \Delta\theta = Ew$ is a bijection between $\mathbb{R}^{m+1}$ and $\mathrm{span}(E) = \mathrm{span}(S)$.

We now rewrite each term in (9) under the substitution $\Delta\theta = Ew$.

Objective. We have

$$g_r^\top \Delta\theta = g_r^\top E w = (E^\top g_r)^\top w = u^\top w,$$

with $u = E^\top g_r$ in the feasible mode.

Constraints. For any enforced constraint $k \in \mathcal{K}$,

$$g_{c_k}^\top \Delta\theta = g_{c_k}^\top E w = (E^\top g_{c_k})^\top w = C_k^\top w.$$

Collecting all such constraints yields $Cw \le \Delta$.

Trust region. Using Lemma 4.1,

$$\tfrac{1}{2}\Delta\theta^\top F \Delta\theta = \tfrac{1}{2}(Ew)^\top F(Ew) = \tfrac{1}{2} w^\top (E^\top F E) w = \tfrac{1}{2}\|w\|_2^2.$$

Thus $\frac{1}{2}\Delta\theta^\top F\Delta\theta \le \delta$ is equivalent to $\|w\|_2 \le R$.

Combining these identities shows that (9) restricted to $\Delta\theta \in \mathrm{span}(S)$ is exactly (17) in the variable w. Optimal solutions correspond under the bijection $\Delta\theta = Ew$. □

A.2.5 Lemma 4.4

Lemma 4.4. *For any $\Delta\theta$ satisfying $\|\Delta\theta\|_F \le R$,*

$$g_{c_k}^\top \Delta\theta \le \nu_k R.$$

Proof. Using $g_{c_k} = F\tilde{g}_{c_k}$,

$$g_{c_k}^\top \Delta\theta = (F\tilde{g}_{c_k})^\top \Delta\theta = \tilde{g}_{c_k}^\top F \Delta\theta = \langle \tilde{g}_{c_k}, \Delta\theta \rangle_F.$$

By Cauchy–Schwarz in the Fisher inner product,

$$\langle \tilde{g}_{c_k}, \Delta\theta \rangle_F \le \|\tilde{g}_{c_k}\|_F \, \|\Delta\theta\|_F.$$

If $\|\Delta\theta\|_F \le R$, then

$$g_{c_k}^\top \Delta\theta \le \|\tilde{g}_{c_k}\|_F \, R.$$

Finally,

$$\|\tilde{g}_{c_k}\|_F^2 = \tilde{g}_{c_k}^\top F \tilde{g}_{c_k} = (O_{\mathrm{full}})_{k+1,k+1} = \nu_k^2,$$

so $\|\tilde{g}_{c_k}\|_F = \nu_k$. □

A.2.6 Proposition 4.5

Proposition 4.5. *If $\Delta_k \ge \nu_k R$, then constraint k cannot be active inside the trust region $\frac{1}{2}\Delta\theta^\top F\Delta\theta \le \delta$ and can be excluded from $\mathcal{K}$ without changing feasibility within that trust region.*

Proof. Inside $\frac{1}{2}\Delta\theta^\top F\Delta\theta \le \delta$, we have $\|\Delta\theta\|_F \le \sqrt{2\delta} = R$. Lemma 4.4 gives, for all such $\Delta\theta$,

$$g_{c_k}^\top \Delta\theta \le \nu_k R.$$

If $\Delta_k \ge \nu_k R$, then $g_{c_k}^\top \Delta\theta \le \Delta_k$ holds automatically for every point in the trust region. Hence constraint k cannot cut the feasible set within the trust region and can be removed without changing feasibility there. □

A.2.7 Lemma 4.6

Lemma 4.6. *Let u_i, u_j be Fisher-unit vectors with $\langle u_i, u_j \rangle_F = \rho \in [-1, 1]$. Then for any $\Delta\theta$ with $\|\Delta\theta\|_F \le R$,*

$$\min_{s \in \{+1,-1\}} \left| \langle u_i, \Delta\theta \rangle_F - s \, \langle u_j, \Delta\theta \rangle_F \right| \le \sqrt{2(1 - |\rho|)}\, R.$$

In particular, if $|\rho| \ge 1 - \varepsilon_{\mathrm{dup}}$, the discrepancy is at most $\sqrt{2\varepsilon_{\mathrm{dup}}}\, R$. One may take $s = +1$ when $\rho \ge 0$ and $s = -1$ when $\rho < 0$.

Proof. Let u_i, u_j satisfy $\|u_i\|_F = \|u_j\|_F = 1$ and $\langle u_i, u_j \rangle_F = \rho \in [-1, 1]$. For any $s \in \{+1, -1\}$,

$$\begin{aligned} \|u_i - s u_j\|_F^2 &= \langle u_i - s u_j,\, u_i - s u_j \rangle_F \\ &= \|u_i\|_F^2 + \|u_j\|_F^2 - 2s \langle u_i, u_j \rangle_F \\ &= 2(1 - s\rho). \end{aligned}$$

Choose $s = +1$ when $\rho \ge 0$ and $s = -1$ when $\rho < 0$, so that $s\rho = |\rho|$ and hence

$$\|u_i - s u_j\|_F = \sqrt{2(1 - |\rho|)}.$$

Then for any $\Delta\theta$,

$$\begin{aligned} \left| \langle u_i, \Delta\theta \rangle_F - s \, \langle u_j, \Delta\theta \rangle_F \right| &= \left| \langle u_i - s u_j, \Delta\theta \rangle_F \right| \\ &\le \|u_i - s u_j\|_F \, \|\Delta\theta\|_F \le \sqrt{2(1 - |\rho|)}\, R, \end{aligned}$$

whenever $\|\Delta\theta\|_F \le R$. Taking the minimum over $s \in \{+1, -1\}$ yields the claim. □

A.2.8 Lemma 4.7

Lemma 4.7. *If $\bar{g}_p \neq 0$ defined by* (27)–(28)*, then for every $i \in \mathcal{V}$,*

$$C_i^\top \bar{g}_p < 0.$$

Proof. Let the violated reduced normals be $\{C_i\}_{i\in\mathcal{V}}$ (rows of C indexed by $\mathcal{V}$). Define the convex hull

$$\mathcal{Q} := \operatorname{conv}\{C_i : i \in \mathcal{V}\} = \left\{ \sum_{i\in\mathcal{V}} \alpha_i C_i : \ \alpha \geq 0, \ \mathbf{1}^\top \alpha = 1 \right\}.$$

Problem (27) computes

$$g^\star = \arg\min_{g\in\mathcal{Q}} \|g\|_2^2, \qquad g^\star = C_\mathcal{V}^\top \alpha^\star, \qquad \bar{g}_p = -g^\star.$$

Since $g^\star$ is the Euclidean projection of 0 onto the closed convex set $\mathcal{Q}$, the projection optimality condition implies that for all $g \in \mathcal{Q}$,

$$(g - g^\star)^\top (0 - g^\star) \leq 0.$$

Rearranging gives

$$g^\top g^\star \geq \|g^\star\|_2^2.$$

Taking $g = C_i$ (note that each $C_i \in \mathcal{Q}$) yields, for every $i \in \mathcal{V}$,

$$C_i^\top g^\star \geq \|g^\star\|_2^2.$$

If $\bar{g}_p \neq 0$, then $g^\star \neq 0$ and $\|g^\star\|_2^2 > 0$. Therefore,

$$C_i^\top \bar{g}_p = -C_i^\top g^\star \leq -\|g^\star\|_2^2 < 0.$$

□

A.3 Reduced QP derivation and KKT solver

Here, we provide a detailed KKT-based solution procedure for the reduced program (17). The presentation parallels the style of the analytical LQCLP derivation in CPO ([1], Thm. 2), but here we treat the *multi-constraint* case and exploit the fact that LOCU's Löwdin basis makes the trust region *Euclidean* (i.e., the quadratic form is I in reduced coordinates). In particular, when $|\mathcal{J}| = 1$, the formulas below reduce exactly to the CPO single-constraint LQCLP solution specialized to $H = I$.

Problem statement. Recall the reduced step subproblem (with $p := m + 1$):

$$\begin{aligned} \max_{w\in\mathbb{R}^p} \quad & u^\top w \\ \text{s.t.} \quad & \|w\|_2 \leq R, \\ & Cw \leq \Delta, \end{aligned} \tag{31}$$

where $C \in \mathbb{R}^{\bar{K}\times p}$ collects the reduced constraint normals and $\Delta \in \mathbb{R}^{\bar{K}}$ collects the corresponding slacks. In LOCU, u is either $E^\top g_r$ (feasible mode) or the recovery direction $\bar{g}_p$ (feasibility-recovery mode), but the solver below is identical.

Convexity and dual attainability. (31) is a linear maximization over a closed convex set (intersection of a Euclidean ball and halfspaces), hence an optimum exists whenever the feasible set is nonempty. In LOCU, nonemptiness is ensured in the recovery mode because we only constrain already-satisfied constraints and thus $w = 0$ is feasible; in the reward mode, $w = 0$ is feasible whenever $\Delta \geq 0$.

A.3.1 KKT conditions

Strong duality and KKT sufficiency. (31) is a convex SOCP (linear objective with a Euclidean ball and halfspace constraints). Whenever the reduced feasible set has a nonempty relative interior (e.g., there exists w such that $\|w\|_2 < R$ and $Cw < \Delta$ for the effective inequality set), strong duality holds and KKT conditions are necessary and sufficient for optimality [6]. If the problem is (nearly) degenerate in practice, our implementation handles it using regularization/tolerances in the active-set routine described below.

Write (31) as a minimization

$$\min_{w} \; -u^\top w \quad \text{s.t.} \quad Cw - \Delta \le 0, \qquad \|w\|_2^2 - R^2 \le 0.$$

The Lagrangian is

$$\mathcal{L}(w, \lambda, \beta) = -u^\top w + \lambda^\top (Cw - \Delta) + \beta(\|w\|_2^2 - R^2),$$

where $\lambda \in \mathbb{R}^{\bar{K}}$ and $\beta \in \mathbb{R}$ satisfy $\lambda \ge 0$, $\beta \ge 0$.

Stationarity:

$$-u + C^\top \lambda + 2\beta w = 0. \tag{32}$$

Primal feasibility:

$$Cw \le \Delta, \qquad \|w\|_2 \le R. \tag{33}$$

Dual feasibility:

$$\lambda \ge 0, \qquad \beta \ge 0. \tag{34}$$

Complementary slackness:

$$\lambda_i \big(C_i^\top w - \Delta_i\big) = 0 \;\; \forall i, \qquad \beta\big(\|w\|_2^2 - R^2\big) = 0. \tag{35}$$

Basic structure. If $u \neq 0$, the optimum typically lies on the ball boundary, i.e., $\|w^\star\|_2 = R$ and thus $\beta^\star > 0$. (If the halfspaces alone already form a bounded polytope that contains the maximizer strictly inside the ball, then $\beta^\star = 0$ is possible; we include this case below for completeness.)

A.3.2 Closed form for a fixed active set

Let $\mathcal{J} \subseteq \{1, \dots, \bar{K}\}$ denote a candidate active set (constraints that are tight at optimum), and write $C_\mathcal{J} \in \mathbb{R}^{|\mathcal{J}| \times p}$ and $\Delta_\mathcal{J} \in \mathbb{R}^{|\mathcal{J}|}$ for the corresponding rows/entries. In KKT form,

$$C_\mathcal{J} w = \Delta_\mathcal{J}, \qquad \lambda_\mathcal{J} \ge 0, \qquad \lambda_{\bar{\mathcal{J}}} = 0.$$

Eliminating w in terms of $(\beta, \lambda_\mathcal{J})$ Assume first that $\beta > 0$. From (32),

$$w(\beta, \lambda_\mathcal{J}) = \frac{1}{2\beta}\Big(u - C_\mathcal{J}^\top \lambda_\mathcal{J}\Big). \tag{36}$$

Imposing the active equalities $C_\mathcal{J} w = \Delta_\mathcal{J}$ yields the reduced linear system

$$\big(C_\mathcal{J} C_\mathcal{J}^\top\big)\lambda_\mathcal{J} = C_\mathcal{J} u - 2\beta\, \Delta_\mathcal{J}. \tag{37}$$

This is exactly the multi-constraint analogue of the scalar equations in CPO's single-constraint LQCLP derivation: there, $|\mathcal{J}| = 1$ and $C_\mathcal{J} C_\mathcal{J}^\top$ is a scalar, leading to a fully analytical solution.

Rank and regularization. If $C_\mathcal{J} C_\mathcal{J}^\top$ is ill-conditioned, we solve

$$\big(C_\mathcal{J} C_\mathcal{J}^\top + \varepsilon I\big)\lambda_\mathcal{J} = C_\mathcal{J} u - 2\beta\, \Delta_\mathcal{J},$$

with a small $\varepsilon > 0$ (or use a pseudoinverse). In LOCU, near-collinear compression makes this case rare.

Algorithm 1 Reduced-space active-set KKT solver for (31)

Require: $u \in \mathbb{R}^p$, $C \in \mathbb{R}^{\bar{K}\times p}$, $\Delta \in \mathbb{R}^{\bar{K}}$, radius $R > 0$, tolerances $\tau_{\text{feas}}, \tau_{\text{dual}}$, regularization $\varepsilon > 0$
Ensure: $w^\star$

If $\|u\|_2 \le \tau_{\text{dual}}$, return $w^\star \leftarrow 0$
$w \leftarrow R\,u/\|u\|_2$ (unconstrained ball maximizer)
if $Cw \le \Delta + \tau_{\text{feas}}\mathbf{1}$ **then**
  return $w^\star \leftarrow w$
end if
Initialize active set $\mathcal{J} \leftarrow \{\arg\max_i (C_i^\top w - \Delta_i)\}$ (most violated)
for $t = 1, 2, \ldots$ until convergence **do**
  Form $C_\mathcal{J}, \Delta_\mathcal{J}$, compute $M_\mathcal{J} = C_\mathcal{J} C_\mathcal{J}^\top$
  Solve $(M_\mathcal{J} + \varepsilon I)\, A_\mathcal{J}^\top = I$ (i.e., $A_\mathcal{J} = (M_\mathcal{J} + \varepsilon I)^{-1}$)
  $w_p \leftarrow C_\mathcal{J}^\top A_\mathcal{J} \Delta_\mathcal{J}$
  $d \leftarrow (I - C_\mathcal{J}^\top A_\mathcal{J} C_\mathcal{J}) u$
  if $\|d\|_2 \le \tau_{\text{dual}}$ **then**
    $w \leftarrow w_p, \quad \beta \leftarrow 0$
  else
    Compute (a, b, c) via (41) and set $t^\star$ by (42)
    $w \leftarrow w_p + t^\star d, \quad \beta \leftarrow 1/(2t^\star)$
  end if
  $\lambda_\mathcal{J} \leftarrow A_\mathcal{J}\big(C_\mathcal{J} u - 2\beta \Delta_\mathcal{J}\big), \quad \lambda_{\bar{\mathcal{J}}} \leftarrow 0$
  if $\min_{j\in\mathcal{J}} \lambda_j < -\tau_{\text{dual}}$ **then**
    Remove $j^\star \in \arg\min_{j\in\mathcal{J}} \lambda_j$ from $\mathcal{J}$ (restore dual feasibility)
    continue
  end if
  if $\max_i (C_i^\top w - \Delta_i) > \tau_{\text{feas}}$ **then**
    Add $i^\star \in \arg\max_i (C_i^\top w - \Delta_i)$ to $\mathcal{J}$ (restore primal feasibility)
    continue
  end if
  return $w^\star \leftarrow w$
end for

Explicit decomposition using a projector Define

$$M_\mathcal{J} := C_\mathcal{J} C_\mathcal{J}^\top \in \mathbb{R}^{|\mathcal{J}|\times|\mathcal{J}|}, \qquad A_\mathcal{J} := M_\mathcal{J}^{-1}, \qquad P_\mathcal{J} := I - C_\mathcal{J}^\top A_\mathcal{J} C_\mathcal{J} \in \mathbb{R}^{p\times p}.$$

$P_\mathcal{J}$ is the (Euclidean) orthogonal projector onto the nullspace of $C_\mathcal{J}$. Also define the minimum-norm particular solution of the active equalities:

$$w_p := C_\mathcal{J}^\top A_\mathcal{J}\, \Delta_\mathcal{J} \quad \text{so that} \quad C_\mathcal{J} w_p = \Delta_\mathcal{J}.$$

Solving (37) and substituting into (36) yields the closed form

$$w(\beta) = w_p + \frac{1}{2\beta} P_\mathcal{J} u. \tag{38}$$

Thus the solution for a fixed $\mathcal{J}$ lies on the affine space $C_\mathcal{J} w = \Delta_\mathcal{J}$ and moves only along the nullspace direction $P_\mathcal{J} u$ as β varies.

Determining β from the ball constraint If $\beta > 0$, complementary slackness enforces $\|w(\beta)\|_2 = R$. Let

$$d := P_\mathcal{J} u, \qquad t := \frac{1}{2\beta}\ (> 0).$$

Then (38) becomes

$$w(t) = w_p + t\, d, \tag{39}$$

and the ball boundary condition is

$$\|w_p + t\, d\|_2^2 = R^2. \tag{40}$$

Expanding gives a quadratic equation in t:

$$a\, t^2 + 2b\, t + c = 0, \qquad a := \|d\|_2^2,\ \ b := w_p^\top d,\ \ c := \|w_p\|_2^2 - R^2. \tag{41}$$

Case 1: $d \neq 0$ **(generic).** Then $a > 0$. Feasibility of the active-equality set inside the ball implies the discriminant is nonnegative:

$$b^2 - ac \geq 0.$$

Among the two roots,

$$t_\pm = \frac{-b \pm \sqrt{b^2 - ac}}{a},$$

the objective increases with t because

$$u^\top w(t) = u^\top w_p + t\, u^\top d = u^\top w_p + t\, \|d\|_2^2,$$

using $u^\top d = u^\top P_\mathcal{J} u = \|P_\mathcal{J} u\|_2^2 = \|d\|_2^2$. Hence the maximizer for the fixed active set uses the *larger* root:

$$t^\star = \frac{-b + \sqrt{b^2 - ac}}{a}, \qquad \beta^\star = \frac{1}{2t^\star}, \qquad w^\star = w_p + t^\star d. \tag{42}$$

Case 2: $d = 0$ **(degenerate/projection case).** Then $P_\mathcal{J} u = 0$, meaning $u \in \mathrm{range}(C_\mathcal{J}^\top)$ and the objective is constant over the affine set $C_\mathcal{J} w = \Delta_\mathcal{J}$: if $u = C_\mathcal{J}^\top y$, then $u^\top w = y^\top C_\mathcal{J} w = y^\top \Delta_\mathcal{J}$. In this case, any feasible w satisfying the active equalities is optimal; we pick the minimum-norm choice $w^\star = w_p$. If $\|w_p\|_2 < R$, one can set $\beta^\star = 0$; if $\|w_p\|_2 = R$, any $\beta^\star \geq 0$ satisfies the ball slackness.

Recovering multipliers $\lambda_\mathcal{J}$ Once $t^\star$ (equivalently $\beta^\star$) is determined, the multipliers on the active constraints follow from (37):

$$\lambda_\mathcal{J}^\star = A_\mathcal{J}\Big(C_\mathcal{J} u - 2\beta^\star \Delta_\mathcal{J}\Big) = A_\mathcal{J}\Big(C_\mathcal{J} u - \frac{1}{t^\star}\Delta_\mathcal{J}\Big), \qquad \lambda_{\bar{\mathcal{J}}}^\star = 0. \tag{43}$$

By construction, $(w^\star, \lambda^\star, \beta^\star)$ satisfies stationarity, the active equalities, and the ball boundary. The remaining KKT requirements are: (i) $\lambda_\mathcal{J}^\star \geq 0$ and (ii) all inactive constraints are satisfied strictly (or at least not violated), i.e., $Cw^\star \leq \Delta$.

A.3.3 Active-set search over $\mathcal{J}$

The closed form above turns the multi-constraint problem into a small-dimensional active-set search: for a candidate $\mathcal{J}$, we can compute a *unique* KKT candidate $(w^\star, \lambda^\star, \beta^\star)$ (up to degenerate cases) via (42) and (43), and then verify feasibility and sign conditions. Because the reduced dimension $p = m + 1$ is small, this procedure is inexpensive.

We use a primal active-set loop: start from the unconstrained ball maximizer, add violated constraints until feasibility is reached, and drop constraints with negative multipliers until dual feasibility is restored.

Remarks. (i) The loop terminates when both primal feasibility ($Cw \leq \Delta$) and dual feasibility ($\lambda \geq 0$) are satisfied; then KKT holds because stationarity and complementarity are enforced by construction. (ii) The only matrix inverse is $(C_\mathcal{J} C_\mathcal{J}^\top + \varepsilon I)^{-1}$ with size $|\mathcal{J}| \leq p$. (iii) When $|\mathcal{J}| = 1$, the computation of $t^\star$ and $\lambda_\mathcal{J}^\star$ reduces to the familiar two-constraint (linear + quadratic) LQCLP algebra in CPO, with the additional simplification that $H = I$ in LOCU's reduced coordinates.

Optional: one-dimensional search for β (implementation alternative) Instead of solving (41) in closed form, one may determine β by a monotone 1D search. For fixed $\mathcal{J}$, (38) implies $\|w(\beta)\|_2$ is nonincreasing in β whenever $d = P_\mathcal{J} u \neq 0$. Therefore, one can bracket β and apply bisection to solve $\|w(\beta)\|_2 = R$. This produces the same $w^\star$ as (42) up to numerical tolerance.

Local approximation error. The preceding LQCLP derivation relies on the local linearization used in CPO [1, Section 6.1, Eq. (11)]. To bound the resulting approximation error, hold the current Fisher matrix F fixed and assume each reward or cost objective J_i, $i \in \{R, C_1, \ldots, C_K\}$, is twice continuously differentiable near the update segment and satisfies

$$\sup_{\tau \in [0,1]} \Big\| F^{-1/2} \nabla_\theta^2 J_i(\pi_{\theta + \tau\Delta\theta}) F^{-1/2} \Big\|_2 \leq L_i, \qquad L_i \geq 0. \tag{44}$$

For any update within the trust region of (9), Taylor's theorem and the fundamental theorem of calculus give

$$\left|J_i(\pi_{\theta+\Delta\theta}) - J_i(\pi_\theta) - \nabla_\theta J_i(\pi_\theta)^\top \Delta\theta\right| \le L_i\delta, \tag{45}$$

$$\left\|F^{-1/2}\big(\nabla_\theta J_i(\pi_{\theta+\Delta\theta}) - \nabla_\theta J_i(\pi_\theta)\big)\right\|_2 \le L_i\sqrt{2\delta}. \tag{46}$$

Consequently, any constraint whose linearized inequality is satisfied obeys

$$g_{c_k}^\top \Delta\theta \le \Delta_k \quad \Longrightarrow \quad J_{C_k}(\pi_{\theta+\Delta\theta}) \le b_k + L_{C_k}\delta. \tag{47}$$

Let ϕ_k be the angle between $F^{-1/2}g_{c_k}$ and $F^{-1/2}\nabla_\theta J_{C_k}(\pi_{\theta+\Delta\theta})$, with ν_k defined in (19). The gradient bound implies

$$\phi_k \le \arcsin\left(\frac{L_{C_k}\sqrt{2\delta}}{\nu_k}\right) \quad \text{if } L_{C_k}\sqrt{2\delta} < \nu_k. \tag{48}$$

Higher local curvature or a smaller original gradient norm weakens this directional bound.

A.4 Practical implementation

Fisher-vector products and natural gradients. Fisher-vector products are computed using Hessian–vector products of the batch-average KL divergence between the cached old policy and the current policy,

$$\bar{D}_{\mathrm{KL}}(\pi_{\theta_{\mathrm{old}}} \,\|\, \pi_\theta) := \mathbb{E}_{s\sim\mathcal{B}}\Big[\mathrm{KL}\big(\pi_{\theta_{\mathrm{old}}}(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid s)\big)\Big], \qquad Fv \approx \nabla_\theta^2\, \bar{D}_{\mathrm{KL}}(\pi_{\theta_{\mathrm{old}}} \,\|\, \pi_\theta)\, v. \tag{49}$$

Natural-gradient directions for the reward and constraints are obtained by conjugate-gradient (CG) solves with Tikhonov damping. We use a fixed number of CG iterations and add a diagonal damping term to the implicit Fisher operator to stabilize the solves under finite-sample noise.

Fisher Gram matrix and Löwdin basis construction. Let S stack the reward direction and the representative constraint directions used for basis construction. We form the Fisher Gram matrix

$$O := S^\top F S \in \mathbb{R}^{(m+1)\times(m+1)}.$$

Before computing the inverse square root, we symmetrize and regularize O as

$$O \leftarrow \tfrac{1}{2}(O + O^\top) + \varepsilon_{\mathrm{reg}} I.$$

We compute $O^{-1/2}$ via eigendecomposition,

$$O = U\Lambda U^\top, \qquad O^{-1/2} = U\Lambda^{-1/2}U^\top,$$

where eigenvalue flooring is applied by replacing each λ_j with $\max\{\lambda_j, \lambda_{\min}\}$ prior to inversion. This prevents numerical instability when the Gram matrix is ill-conditioned. The Löwdin basis is then constructed as

$$E := S\, O^{-1/2}.$$

Reduced quantities and parameter update. Reduced-space quantities are obtained by projection onto the Löwdin basis:

$$u := E^\top g_r, \qquad C_k^\top := E^\top g_{c_k}, \quad k \in \mathcal{K}.$$

The trust-region radius is set to $R = \sqrt{2\delta}$. After solving the reduced program for $w^\star$, the actor update in parameter space is given by

$$\Delta\theta^\star = E w^\star.$$

Reward-aligned constraint filtering (optional). During basis construction, constraints whose Fisher cosine similarity with the reward direction exceeds a fixed threshold are optionally excluded from basis candidates. This heuristic avoids introducing nearly collinear reward–constraint directions into the reduced basis. In our benchmark environments, strong alignment between reward and constraint gradients is rarely observed in practice, and we found that disabling this filter does not measurably affect performance. Accordingly, we retain this step as an optional numerical safeguard rather than a core assumption of the method.

Algorithm 2 LOCU one actor update

Require: policy π_θ, trust region δ, limits $\{b_k\}_{k=1}^K$, thresholds $\varepsilon_{\mathrm{nz}}, \varepsilon_{\mathrm{dup}}$, regularization $\varepsilon_{\mathrm{reg}}$

Estimate reward and cost gradients g_r, $\{g_{c_k}\}_{k=1}^K$
Estimate Fisher-vector products for the FIM F
Compute slacks $\Delta_k = b_k - J_{C_k}(\pi_\theta)$ and set $R \leftarrow \sqrt{2\delta}$
Compute natural gradients $\tilde{g}_r = F^{-1} g_r$ and $\tilde{g}_{c_k} = F^{-1} g_{c_k}$ (with damping if needed)
Form $S_{\mathrm{full}} = [\tilde{g}_r, \tilde{g}_{c_1}, \ldots, \tilde{g}_{c_K}]$ and $O_{\mathrm{full}} = S_{\mathrm{full}}^\top F S_{\mathrm{full}}$
Compute $\nu_k, \mu_k, \varrho_{ij}$ via (19)–(20)
Build enforced set $\mathcal{K}$ by excluding (i) non-binding constraints (Prop. 4.5) and (ii) near-zero constraints $\nu_k \le \varepsilon_{\mathrm{nz}}$
Build near-collinear groups on $\mathcal{K}$ using edges $|\varrho_{ij}| \ge 1 - \varepsilon_{\mathrm{dup}}$
Set $i_\ell \in \arg\min_{k \in \mathcal{G}_\ell} \mu_k$ for each group $\mathcal{G}_\ell$
Form $S = [\tilde{g}_r, \tilde{g}_{c_{i_1}}, \ldots, \tilde{g}_{c_{i_m}}]$ and $O = S^\top F S$; regularize O if needed
Compute Löwdin basis $E = S O^{-1/2}$
Build reduced quantities $u = E^\top g_r$ and $C_k^\top = E^\top g_{c_k}$ for all $k \in \mathcal{K}$
if all retained slacks satisfy $\Delta_k \ge 0$ **then**
    Solve the reduced program (17) for $w^\star$
else
    Compute Pareto descent direction $\bar{g}_p$ via (27)–(28)
    Set $u \leftarrow \bar{g}_p$ and solve the recovery program (30) for $w^\star$
end if
Set $\Delta\theta^\star \leftarrow E w^\star$
Update $\theta \leftarrow \theta + \Delta\theta^\star$

Numerical tolerances. All linear solves in the reduced space use small ridge regularization. Feasibility and dual checks in the active-set routine are evaluated with fixed tolerances. If the reduced objective norm falls below a small threshold, the update defaults to the zero step.

B Experimental Details

B.1 Task details

B.1.1 Circle Task from Bullet–safety–gym

We use the Circle tasks from Bullet-Safety-Gym [8] with Ball, Car, and Drone agents and the multi-constraint extension from prior work [29]. The reward of Circle task is defined as

$$r_t = 0.1 \, \frac{p_t^\top (J v_t)}{1 + \big| \|p_t\|_2 - r_{\mathrm{circ}} \big|} + 0.01 \, r_t^{\mathrm{spec}}, \qquad J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}. \tag{50}$$

This term captures the signed tangential velocity around the origin, yielding positive values for clockwise motion along the circular trajectory. The normalization factor attenuates the reward as the agent deviates from the target radius r_{circ}, so the reward is maximized near $\|p_t\|_2 \approx r_{\mathrm{circ}}$ under sustained clockwise rotation.

The agent-specific reward component r_t^{spec} follows

$$r_t^{\mathrm{spec}} = \begin{cases} -0.5 \, \|a_t\|_2, & \text{Ball}, \\ 0, & \text{Car}, \\ r_t^{\mathrm{alive}} - 0.01 \, \|\omega_t\|_2^2 - c_t^{\mathrm{ground}}, & \text{Drone}. \end{cases} \tag{51}$$

Here a_t denotes the clipped action and ω_t the angular velocity. The term c_t^{ground} denotes the ground-contact penalty used by the environment. This shaping term provides mild agent-dependent regularization without altering the task objective. For the Ball agent, the ℓ_2 action penalty regularizes control effort. The Car agent uses no additional shaping beyond the task reward. For the Drone agent, r_t^{alive} provides a positive reward when the agent remains in a valid flying state and a negative reward otherwise, while the remaining terms discourage aggressive rotations and unsafe ground contacts.

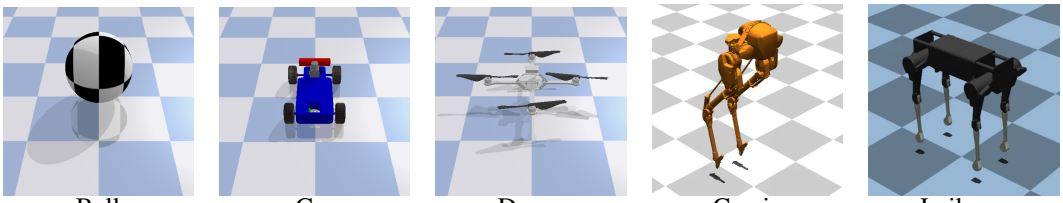


Figure 5: Agents used in the multi-constraint evaluation. Ball, Car, and Drone are BulletSafetyGym agents, while Cassie and Laikago are bipedal and quadrupedal robots, respectively.

Table 1: BC-v3 stress-test configurations used in Figure 2.

Setting	Low-velocity coefficients κ_{lo}	Boundary thresholds x_{bnd}	High-velocity coefficients κ_{hi}
Base (v3)	$\{\mathbf{0.6}^{(-a)}\}$	$\{\mathbf{6.0}^{(-a)}\}$	$\{\mathbf{1.0}^{(-a)}\}$
Redundant-2	$\{\mathbf{0.6}^{(-a)}, 0.65\}$	$\{\mathbf{6.0}^{(-a)}, 6.5\}$	$\{\mathbf{1.0}^{(-a)}, 0.95\}$
Redundant-3	$\{\mathbf{0.6}^{(-a)}, 0.65, 0.7\}$	$\{\mathbf{6.0}^{(-a)}, 6.5, 7.0\}$	$\{\mathbf{1.0}^{(-a)}, 0.95, 0.9\}$

Constraints. We use three per-step binary costs: a boundary cost, a high-velocity cost, and a low-velocity cost. Let x_t denote the agent position along the x-axis and v_t its planar speed. With boundary threshold $x_{\text{bnd}} > 0$, base velocity limit $v_{\max}$, and scaling coefficients $\kappa_{\text{hi}} \in (0, 1]$ and $\kappa_{\text{lo}} \in (0, 1)$, we define

$$c_{\text{bdry},t} = \mathbb{I}\{|x_t| > x_{\text{bnd}}\}, \tag{52}$$

$$c_{\text{hi},t} = \mathbb{I}\{v_t > \kappa_{\text{hi}}\, v_{\max}\}, \tag{53}$$

$$c_{\text{lo},t} = \mathbb{I}\{v_t < \kappa_{\text{lo}}\, v_{\max}\}. \tag{54}$$

The two-constraint setting $\{c_{\text{bdry}}, c_{\text{hi}}\}$ is referred to as **v2**, and the three-constraint setting $\{c_{\text{bdry}}, c_{\text{hi}}, c_{\text{lo}}\}$ as **v3**. Unless otherwise stated, the base v3 configuration uses $x_{\text{bnd}} = 6.0$, $\kappa_{\text{hi}} = 1.0$, and $\kappa_{\text{lo}} = 0.6$.

BC-v3 Variants. Figure 2 reports results on several BC-v3 variants obtained by tightening and duplicating constraints on top of the base v3 setting. All variants share the same task dynamics and reward, and only differ in the constraint coefficients summarized in Table 1. Boldfaced coefficients (marked with $^{(-a)}$) correspond to the constraints denoted as "–a" in the results and learning-curve figures.

As redundancy increases, the low-velocity coefficient κ_{lo} increases, raising the minimum admissible speed. This tightens the low-velocity constraint and prolongs the recovery phase early in training, since the agent must first escape low-speed states before achieving sustained motion along the circle. In parallel, the high-velocity coefficient κ_{hi} decreases, tightening the upper speed limit. Because the task reward is proportional to tangential velocity, this directly reduces the achievable return by limiting the speed at which the agent can traverse the trajectory.

The tightest boundary coefficient remains fixed across all variants. Therefore, additional boundary constraints do not further restrict the feasible set, but only introduce near-collinear constraint directions, isolating the effect of constraint redundancy on optimization behavior.

B.1.2 Locomotion tasks

In locomotion tasks the agent tracks velocity commands on flat terrain. At the beginning of each episode, a command vector

$$(v_x^{\text{cmd}}, v_y^{\text{cmd}}, \omega_z^{\text{cmd}})$$

is sampled, where v_y^{cmd} is fixed to 0 to simplify the task. In the implementation, $(v_x^{\text{cmd}}, \omega_z^{\text{cmd}})$ are sampled uniformly from robot-specific ranges: Cassie uses $v_x^{\text{cmd}} \sim \mathcal{U}[-1, 1]$, while Laikago uses $v_x^{\text{cmd}} \sim \mathcal{U}[-1, 2]$, and both use $\omega_z^{\text{cmd}} \sim \mathcal{U}[-0.5, 0.5]$. The reward penalizes squared tracking error between commanded and realized base velocities and adds a small power regularization term.

Constraint Setting At time step t, let θ_t denote the angle between the robot base z-axis and the world z-axis, and let $a > 0$ be a fixed balance threshold. Let h_t denote the center-of-mass (CoM) height with a robot-specific minimum threshold $h_{\min} > 0$. Let $\hat{\xi}_t \in \{-1, 1\}^{n_{\text{legs}}}$ be the measured foot contact vector and $\xi_t \in \{-1, 1\}^{n_{\text{legs}}}$ the desired periodic contact pattern derived from phase variables.

We define the following per-step costs:

$$c_{\text{bal},t} = \mathbb{I}\{\theta_t > a\}, \tag{55}$$

$$c_{\text{hgt},t} = \mathbb{I}\{h_t < h_{\min}\}, \tag{56}$$

$$c_{\text{foot},t} = \frac{1}{n_{\text{legs}}} \sum_{i=1}^{n_{\text{legs}}} \mathbb{I}\{\hat{\xi}_{t,i} \neq \xi_{t,i}\}. \tag{57}$$

For the balance constraint, we use $a = 15°$. For the height constraint, we set $h_{\min} = 0.35$ for Laikago and $h_{\min} = 0.7$ for Cassie. The desired contact pattern ξ_t is defined using per-leg phase variables with a fixed frequency, following the benchmark specification.

Constraint limits are computed from target per-step cost rates by scaling with $1/(1-\gamma)$, following the standard rule for per-step costs bounded in $[0, 1]$. The foot timing constraint is assigned a looser limit than the balance and height constraints, as it primarily serves as a motion regularizer rather than a strict safety constraint.

Episode length and termination handling. Episodes run for a fixed horizon of T steps and are not terminated early upon falling. When a fall is detected, the state and reward are held fixed until the horizon ends, preventing termination effects from confounding constraint evaluation.

B.1.3 Bipedal Navigation task

Objective. To increase task diversity beyond velocity-tracking locomotion, we construct a Cassie goal-navigation task with obstacle avoidance, denoted as CassieGoalHazard-v0. The task extends the Cassie locomotion environment by replacing the commanded-velocity reward with a planar goal-reaching objective and by adding hazard-avoidance constraints adapted from Safety-Gymnasium [10]. This setting is designed to couple legged locomotion stability with state-dependent navigation constraints while preserving a feasible but nontrivial multi-constraint control problem.

At the beginning of each episode, eight circular hazards and one goal are sampled in a square map with extents $[-3.5, 3.5] \times [-3.5, 3.5]$. The goal radius is $r_{\text{goal}} = 0.5$, and each hazard has radius $r_{\text{haz}} = 0.35$. The observation augments the Cassie proprioceptive state with 16-bin pseudo-LiDAR measurements for the goal and hazards, together with the goal distance.

Let d_t denote the planar distance between the robot base and the current goal at time step t. The reward is the distance reduction toward the goal plus a sparse goal bonus:

$$r_t = d_{t-1} - d_t + 0.2\,\mathbb{I}\{d_t \leq r_{\text{goal}}\}. \tag{58}$$

When the goal is reached, a new goal is sampled while the hazard layout remains fixed. Thus, the task rewards repeated navigation progress rather than a single terminal success event.

Constraint setting. The task retains the three Cassie locomotion-stability costs from Section B.1.2 and adds three navigation-specific costs. The full cost vector is ordered as$(c_{\text{bal}}, c_{\text{hgt}}, c_{\text{foot}}, c_{\text{spd}}, c_{\text{haz}}, c_{\text{eng}})$. The first three costs are the balance, height, and foot-timing costs used in the locomotion task, with the Cassie height threshold set to $h_{\min} = 0.7$.

For the additional costs, let $v_{x,t}$ denote the forward base velocity, let $\mathcal{P}_t$ be the set of floor-contact points of the robot at time t, and let $\mathcal{H}$ be the set of hazard centers. We define

$$c_{\text{spd},t} = \mathbb{I}\{|v_{x,t}| > 1.0\}, \tag{59}$$

$$c_{\text{haz},t} = \mathbb{I}\{\exists p \in \mathcal{P}_t, \exists h \in \mathcal{H} : \|p - h\|_2 \leq r_{\text{haz}}\}, \tag{60}$$

$$c_{\text{eng},t} = \mathbb{I}\{e_t > 13.0\}, \tag{61}$$

where e_t is the average actuator-power proxy computed from the absolute actuator force times joint velocity over simulation substeps.

Table 2: Actor/critic architectures and training hyperparameters (shared across methods unless noted).

Parameter	LOCU	SDAC(A)	PCRPO / CRPO	TRPOL / GS	PPOL / GS
Actor MLP (hidden, act.)	(128, 128) ReLU / (512, 512) LeakyReLU				(128, 128) ReLU / –
Policy distribution	Diag. Gaussian / squashed Gaussian				Diag. Gaussian / –
Action bounding	clip / tanh-squash				clip / –
Critic architecture	scalar V-critic / distributional Q-critic				scalar V-critic / –
Distributional atoms (M/M')	– / (25/50)				– / –
Target KL	0.001				0.02 / –
Critic optimization iters	20 / 40				–
Critic learning rate	1e-4 / 3e-4				1e-4 / –
L2 regularization	0.001				–
Damping coefficient	0.1 / 0.01				–
CG iters	10				–
Batch size (rollout)	10000				
Minibatch size	–				256 / –
Discount factor γ	0.99				
GAE / trace-decay λ	0.95 / 0.97				
Advantage normalization	True				
PID gains (k_p, k_i, k_d)	–	–	–	(0.2, 0.002, 0.1)	(0.15, 0.005, 0.1)

Entries a/b denote Bullet-Safety-Gym / Locomotion, respectively.
"–" indicates not applicable or not evaluated; PPOL/GS was not run on locomotion tasks.

Constraint limits are converted by the same $1/(1-\gamma)$ scaling used for the locomotion tasks. These thresholds were selected through pilot calibration to keep simultaneous satisfaction feasible while avoiding overly loose constraints.

This task introduces qualitatively heterogeneous constraints. Goal-reaching encourages faster progress, while hazard avoidance can require detours, speed regulation can restrict aggressive motion, and energy control penalizes high-actuation strategies. The resulting environment therefore tests whether an algorithm can recover feasibility from infeasible initialization while still incorporating reward-improving directions once the active constraints are controlled. All experiments use the same fixed horizon of $T = 500$ steps.

B.2 Implementation details

B.2.1 Baseline Implementation

SDAC, PID Lagrangian, and Gradient Shaping (GS). For SDAC, we use the official implementation released by the original authors. For PID Lagrangian and Gradient Shaping (GS), we adopt the implementation built on top of the FSRL framework [16], as released by the authors of GS. Across these baselines, we align common experimental settings and remaining algorithm-specific components, including multiplier dynamics, and shaping rules, are kept identical to the respective authors' implementations without modification.

CRPO. We build CRPO on top of the CRPO implementation provided by OmniSafe [11] and extend it to the multi-constraint setting. When multiple constraints are violated simultaneously, we sample a single constraint to update the policy at each iteration. The sampling probability is proportional to the magnitude of constraint violation, so that more severely violated constraints are prioritized during training.

PCRPO. To the best of our knowledge, no official public implementation of PCRPO is available, although the original paper reports that PCRPO is implemented and evaluated using the OmniSafe framework. We therefore implement PCRPO directly following the algorithmic description and the configuration settings reported in the paper. In particular, the appendix provides a configuration table that lists shared hyperparameter settings for PCRPO, including the use of CG iterations. Based on this specification, we implement PCRPO as a second-order primal method that computes natural-gradient style updates using a CG with its soft-switching method.

PCRPO uses two slack thresholds per constraint, denoted as an upper slack and a lower slack, to determine which objective is optimized at each update. We set these thresholds by scaling the corresponding constraint limit b_k. In Bullet-Safety-Gym, we use an upper slack of $1.25\, b_k$ and a lower slack of $0.75\, b_k$. In locomotion tasks, we use an upper slack of $1.0\, b_k$ and a lower slack of $0.75\, b_k$, applied per constraint.

At each iteration, constraints are categorized into three groups based on their current costs relative to the slack thresholds: (i) constraints with cost above the upper slack, (ii) constraints with cost between the upper and lower slacks, and (iii) constraints with cost below the lower slack. If any constraint falls into group (i), one violated constraint is sampled from this group and a constraint-reduction update is performed. Otherwise, if group (ii) is non-empty, one constraint is sampled and a reward update is applied together with the gradient-manipulation rule specified by PCRPO to mitigate interference between reward and constraint gradients. If all constraints belong to group (iii), a feasible update is performed.

Clarabel. We use Clarabel to solve the local constrained policy-update problem in two forms: full QP and reduced QP. The full QP optimizes the coefficients of the natural reward and cost gradients, while the reduced QP optimizes the update in LOCU's Löwdin coordinates, including the orthogonal reward direction. For constraint recovery, the full QP minimizes total constraint slack when the reward-maximization problem is infeasible. The reduced QP minimizes slack for currently violated constraints while enforcing the other constraints.

B.2.2 Mean–std constraint and distributional cost critics

SDAC adopts the mean–std safety measure $F(Z;\alpha_{\mathrm{lo}}) = \mathbb{E}[Z] + \frac{\phi(\Phi^{-1}(\alpha_{\mathrm{lo}}))}{\alpha_{\mathrm{lo}}}\mathrm{Std}[Z]$ motivated as a practical surrogate via the Gaussian–CVaR connection. Following the discussion in SDAC, we treat mean–std as an uncertainty-weighted surrogate rather than an exact CVaR constraint [14].

For locomotion tasks, we use a distributional cost critic and denote the predicted quantile samples for constraint k at (s,a) by $\{z_k^{(m)}(s,a)\}_{m=1}^{M}$. In our locomotion experiments, we adopt the same task definitions and training protocol as SRCPO [15]. We keep the distributional critic training unchanged, including the quantile regression objective and the construction of bootstrapped target quantiles, and modify only the aggregation used to compute the mean–std surrogate from predicted quantiles.

The original SDAC aggregation estimates the mean–std surrogate by mixing quantile samples across the batch and then taking a single global standard deviation:

$$\begin{aligned} \widehat{F}_k^{\mathrm{mixed}} &= \mathrm{Mean}\Big(\{z_k^{(m)}(s,a)\}_{(s,a)\in\mathcal{B},\,m}\Big) + \sigma(\alpha_{\mathrm{lo}})\mathrm{Std}\Big(\{z_k^{(m)}(s,a)\}_{(s,a)\in\mathcal{B},\,m}\Big), \\ \sigma(\alpha_{\mathrm{lo}}) &= \frac{\phi(\Phi^{-1}(\alpha_{\mathrm{lo}}))}{\alpha_{\mathrm{lo}}}. \end{aligned} \tag{62}$$

In our locomotion experiments, this batch mixed aggregation can yield unstable updates and, in some runs, prevent convergence. We therefore compute the statistics per state–action pair first and then average across the batch:

$$\widehat{F}_k^{\mathrm{cond}} = \mathrm{Mean}_{(s,a)\in\mathcal{B}}\Big(\mathrm{Mean}_m[z_k^{(m)}(s,a)] + \sigma(\alpha_{\mathrm{lo}})\,\mathrm{Std}_m[z_k^{(m)}(s,a)]\Big). \tag{63}$$

This modification reduces overly conservative behavior caused by batch heterogeneity by ensuring that the standard deviation term reflects within-sample distributional uncertainty rather than variation across states within the batch.

C Supplementary Analysis and Results

C.1 Multi-constraint update behaviors of existing Primal-dual algorithms

We summarize multi-constraint update behaviors in representative safe policy optimization algorithms by grouping methods by how they incorporate multiple constraints into a single actor step: (i) *sequential constraint projection* onto linearized halfspaces and (ii) *scalarized constraint handling* via a weighted aggregation of constraint signals. We additionally discuss practical *hyperparameter tuning difficulties* that arise in multi-constraint settings.

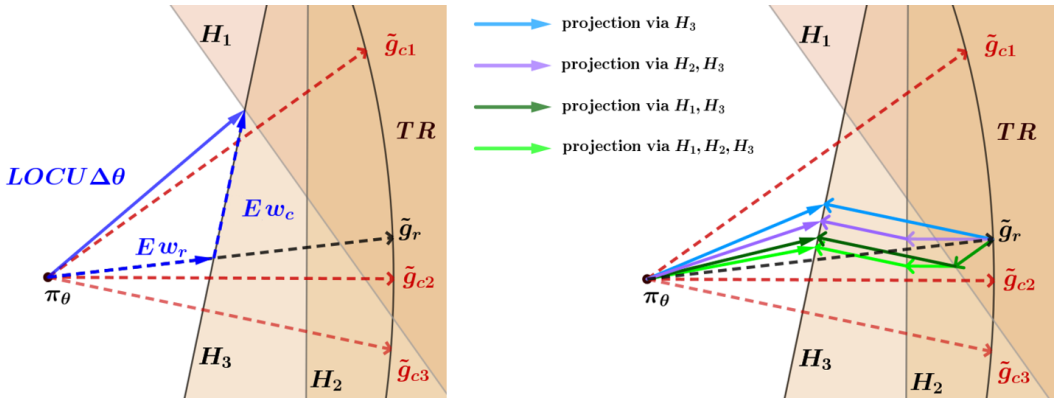


Figure 6: Geometric comparison between LOCU and PCPO updates. LOCU solves a reduced QP and produces an order-invariant update under the local model. PCPO applies sequential projections onto constraint halfspaces, which can produce order-dependent update directions.

C.1.1 Sequential constraint projection

A common projection-based approach is to first compute a reward-improving step and then enforce constraint satisfaction by projecting it onto a linearized constraint halfspace under a chosen metric. A representative instance is PCPO [28], which in the single-constraint case solves the projection

$$\Delta\theta^+ = \arg\min_{\Delta\theta\in\mathbb{R}^n} \frac{1}{2}(\Delta\theta - \Delta\theta^{(0)})^\top L\,(\Delta\theta - \Delta\theta^{(0)}) \quad \text{s.t.} \quad g_c^\top \Delta\theta \le \Delta, \tag{64}$$

where L is the projection metric. PCPO considers $L = I$ for ℓ_2 projection and $L = F$ for Fisher-metric projection. The solution is the closed-form correction along $L^{-1}g_c$:

$$\Delta\theta^+ = \Delta\theta^{(0)} - \alpha\, L^{-1}g_c, \qquad \alpha = \max\left\{0,\ \frac{g_c^\top \Delta\theta^{(0)} - \Delta}{g_c^\top L^{-1} g_c}\right\}. \tag{65}$$

For $L = F$, define $u_k = \tilde{g}_{c_k} / \left\|\tilde{g}_{c_k}\right\|_F$ and $\mu_k = \Delta_k / \left\|\tilde{g}_{c_k}\right\|_F$. The Fisher-metric projection onto the halfspace $\langle u_k, \Delta\theta\rangle_F \le \mu_k$ is

$$P_k(x) := \arg\min_{y:\,\langle u_k, y\rangle_F \le \mu_k} \frac{1}{2}\left\|y - x\right\|_F^2 = x - \left[\langle u_k, x\rangle_F - \mu_k\right]_+ u_k, \tag{66}$$

where $[t]_+ := \max\{t, 0\}$.

As discussed in PCPO [28], in the multi-constraint setting PCPO can apply these halfspace projections sequentially across constraints. However, this sequential projection procedure can exhibit order dependence and step shrink when constraint normals are correlated. Figure 6 visualizes the local model and contrasts LOCU with sequential-projection updates. The left panel corresponds to LOCU's single reduced-space solve. The right panel depicts sequential halfspace projections, where order dependence is shown in (68)–(69) and step shrink is quantified by (70).

Consider Fisher halfspaces $\mathcal{H}_k := \{\Delta\theta : \langle u_k, \Delta\theta\rangle_F \le \mu_k\}$. Starting from the reward-only step $\Delta\theta^{(0)}$, a sequential projection update applies

$$\Delta\theta^{(t)} = P_{\sigma_t}\left(\Delta\theta^{(t-1)}\right), \qquad t = 1, 2, \ldots, m, \tag{67}$$

for some ordering σ.

For two constraints i and j, define

$$s_i := \langle u_i, x\rangle_F, \quad s_j := \langle u_j, x\rangle_F, \quad t_i := [s_i - \mu_i]_+, \quad t_j := [s_j - \mu_j]_+, \quad \rho_{ij} := \langle u_i, u_j\rangle_F.$$

Then

$$P_j(P_i(x)) = x - t_i u_i - \Big[\, s_j - \rho_{ij} t_i - \mu_j \,\Big]_+ u_j, \tag{68}$$

$$P_i(P_j(x)) = x - t_j u_j - \Big[\, s_i - \rho_{ij} t_j - \mu_i \,\Big]_+ u_i. \tag{69}$$

If both constraints are active and $\rho_{ij} \neq 0$, the correction magnitudes differ and the final update depends on the order.

Sequential projections also reduce the Fisher step length. Let $t_k := [\langle u_k, x\rangle_F - \mu_k]_+$. Using $\|u_k\|_F = 1$,

$$\|P_k(x)\|_F^2 = \|x\|_F^2 - t_k\,(t_k + 2\mu_k). \tag{70}$$

When $\mu_k \geq 0$, any active projection $t_k > 0$ strictly decreases $\|\cdot\|_F$, so $\left\|\Delta\theta^{(t)}\right\|_F$ is non-increasing along (67).

The reward improvement in the local model is attenuated in proportion to Fisher correlations. Let $u_r = \tilde{g}_r / \|\tilde{g}_r\|_F$. Then

$$\langle u_r, P_k(x)\rangle_F = \langle u_r, x\rangle_F - t_k\,\langle u_r, u_k\rangle_F. \tag{71}$$

If $\langle u_r, u_k\rangle_F$ is not small across multiple active constraints, sequential projections repeatedly subtract reward-aligned components and can produce overly conservative updates.

C.1.2 Scalarized constraint handling

Another prevalent design is to reduce a vector-valued constraint problem to a *single scalar* surrogate by aggregating multiple constraint signals through a weighted sum (typically via dual variables). This avoids explicit geometric reasoning about how constraint normals interact; instead, interactions appear only through the aggregate scalar and the resulting single update direction.

Constrained Update Projection (CUP) CUP [27] explicitly separates an intermediate performance-improvement iterate $\pi_{\theta_{k+\frac{1}{2}}}$ from a *projection-to-safety* step. In the projection stage, CUP solves a constrained KL projection in policy space:

$$\pi_{\theta_{k+1}} = \arg\min_{\pi_\theta \in \Pi_\theta} \bar{D}_{\mathrm{KL}}\left(\pi_{\theta_{k+\frac{1}{2}}}, \pi_\theta\right) \quad \text{s.t.} \quad \hat{C}_{\pi_{\theta_k}}(\pi_\theta, \beta_k) \leq b, \tag{72}$$

where $\hat{C}$ denotes a surrogate cost and β_k is a penalty coefficient ensuring the theoretical safety bound. The optimization is implemented via a primal–dual routine with Lagrange multipliers associated with constraints. In the multi-constraint case, multipliers are updated per-constraint based on their individual violations, and the resulting policy update is driven by the aggregated penalty direction induced by the current multiplier vector.

First Order Constrainted Optimization in Policy Space (FOCOPS) FOCOPS first solves a constrained optimization in policy space with a KL trust region, obtaining a non-parametric optimal policy (shown in [30], Thm. 1):

$$\pi^*(a \mid s) \propto \pi_{\theta_k}(a \mid s)\,\exp\left(\frac{1}{\lambda}\left(A_R^{\pi_{\theta_k}}(s,a) - \sum_k \nu_k A_{C_k}^{\pi_{\theta_k}}(s,a)\right)\right), \tag{73}$$

where $\lambda > 0$ and $\{\nu_k \geq 0\}$ are dual variables. FOCOPS then projects back to the parametric class by minimizing KL:

$$\theta_{k+1} = \arg\min_\theta\, \mathbb{E}_{s\sim d^{\pi_{\theta_k}}}\left[\mathrm{KL}(\pi^*(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid s))\right]. \tag{74}$$

Overlap and cancellation under scalarization. Both CUP and FOCOPS rely on a scalarized aggregate of multiple constraints, e.g., the aggregated advantage in (73)

$$\hat{A}(s,a) := A_R(s,a) - \sum_{k=1}^{K} \nu_k A_{C_k}(s,a),$$

or, in parameter space, an aggregated penalty direction of the form

$$\tilde{g}_{\text{eff}} = \tilde{g}_r - \sum_{k=1}^{K} \lambda_k \tilde{g}_{c_k}.$$

Table 3: PID coefficient candidate values used in our experiments.

Coefficient	Candidate values
K_P	$\{0.25, 1\}$
K_I	$\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$
K_D	$\{0.01, 0.1, 1\}$

In multi-constraint regimes, the behavior of the aggregate is governed by pairwise correlations among constraint directions. In Fisher geometry, the penalty direction satisfies

$$\left\| \sum_{k=1}^{K} \lambda_k \tilde{g}_{c_k} \right\|_F^2 = \sum_{i=1}^{K} \lambda_i^2 \, \|\tilde{g}_{c_i}\|_F^2 \; + \; 2 \sum_{1 \le i < j \le K} \lambda_i \lambda_j \, \langle \tilde{g}_{c_i}, \tilde{g}_{c_j} \rangle_F. \tag{75}$$

The cross terms encode *interference*: if $\langle \tilde{g}_{c_i}, \tilde{g}_{c_j} \rangle_F < 0$ (opposing directions), the aggregate can exhibit partial cancellation, weakening constraint correction even when individual constraints are violated; if $\langle \tilde{g}_{c_i}, \tilde{g}_{c_j} \rangle_F > 0$ (overlapping directions), the aggregate can over-emphasize a redundant direction, effectively reducing the rank of meaningful constraint information and making the update sensitive to the instantaneous multiplier vector. Since scalarization does not explicitly model the constraint geometry, trade-offs among conflicting constraints are resolved implicitly through multiplier weights rather than through an explicit reduced-space analysis of active constraints.

C.1.3 Hyperparameter tuning difficulties

Primal–dual Lagrangian methods handle constraints by introducing multipliers and optimizing

$$\max_{\theta} \min_{\lambda \ge 0} \; J_R(\pi_\theta) \; - \; \sum_{k=1}^{K} \lambda_k \big(J_{C_k}(\pi_\theta) - b_k \big), \tag{76}$$

so the actor update direction is $g_r - \sum_k \lambda_k g_{c_k}$. In practice, the training behavior can be dominated by the dual feedback loop and becomes sensitive to the initialization of λ and the relative step sizes of primal and dual updates, which often leads to substantial hyperparameter tuning and unstable learning [19, 13].

PID Lagrangian replaces simple dual ascent with a PID-style controller over the constraint error $e_k := J_{C_k}(\pi_\theta) - b_k$ [23],

$$\lambda_k \leftarrow \Big[\lambda_k + K_P e_k + K_I \sum_{\tau \le t} e_{k,\tau} + K_D (e_k - e_{k-1}) \Big]_+, \tag{77}$$

However, selecting K_P, K_I, and K_D remains a practical bottleneck.

We evaluated PID PPO-Lagrangian with gradient shaping [29] under different PID coefficient choices. We used similar PID coefficient sets used for the coefficient learning-curve comparisons in Appendix C of the PID-Lagrangian paper [23], summarized in Table 3. All combinations were evaluated with five random seeds. In Figure 7, each curve was obtained by averaging all runs that shared the coefficient value of interest over the remaining two PID coefficients and seeds, and the shaded region denotes the standard deviation over the same set of runs.

Gradient shaping mitigates cancellation and overlap among constraint gradients in the actor update [29], yet sensitivity of the multiplier dynamics from (77) still remains a challenge. Across the tested settings, only a narrow regime such as $K_I = 10^{-3}$ yields relatively stable boundary tracking in this task, suggesting that the robustness reported in single-constraint PID Lagrangian analyses [23] can be weakened when multiple constraints are coupled through a policy update.

C.2 Perturbation sensitivity of Löwdin and Gram-Schmidt coefficients

We compare the perturbation sensitivity of Löwdin and Gram–Schmidt orthogonalization through theoretical analysis and empirical evaluation, focusing on how stochastic noise propagates through their respective orthogonalization procedures.

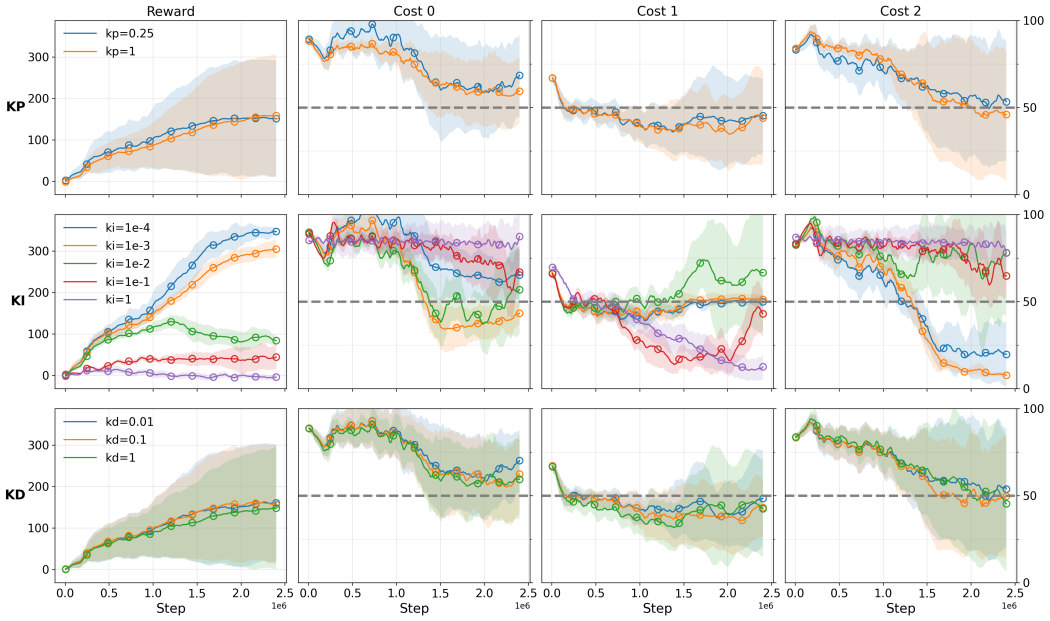


Figure 7: Learning-curve comparison of PID PPO-Lagrangian under P I D coefficient variations on the Bullet Safety Gym BC-v3 task. Dashed lines in the cost plots indicate the corresponding limits.

C.2.1 Theoretical analysis

Using the notation in Sections 3.2, 3.3, and 4.1, let $S = [\tilde{g}_r, \tilde{g}_{c_{i_1}}, \ldots, \tilde{g}_{c_{i_m}}]$ and $O = S^\top F S$. Assume $O \succ 0$, possibly with small eigenvalues due to near-collinear directions. Let $\widehat{S} := S + \Xi$ and $\widehat{O} := \widehat{S}^\top F \widehat{S}$ for stochastic gradient noise Ξ small enough that $\widehat{O} \succ 0$. Then

$$\Delta O := \widehat{O} - O = S^\top F \Xi + \Xi^\top F S + \Xi^\top F \Xi. \tag{78}$$

Its norm is bounded by

$$\|\Delta O\|_2 \leq 2\|F^{1/2} S\|_2 \|F^{1/2} \Xi\|_2 + \|F^{1/2} \Xi\|_2^2. \tag{79}$$

For LOCU, since Proposition 4.2 gives $E^\top g_r = O^{1/2} e_1$ and $E^\top g_{c_{i_j}} = O^{1/2} e_{j+1}$, the sensitivity depends on the noise of the square root.

For Löwdin Orthogonalization. Let $O = Q \Lambda Q^\top$, with Q the eigenvector matrix so that $Q^\top(\cdot)Q$ represents a change to the eigenbasis, and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_{m+1})$. Then, defining $\Delta_L := \widehat{O}^{1/2} - O^{1/2}$, the first-order term satisfies the Sylvester equation:

$$O^{1/2} \Delta_L + \Delta_L O^{1/2} = \Delta O + \mathcal{O}(\|\Delta O\|_2^2). \tag{80}$$

So in the eigenbasis of O,

$$Q^\top \Delta_L Q = \left[\frac{(Q^\top \Delta O Q)_{ab}}{\sqrt{\lambda_a} + \sqrt{\lambda_b}} \right]_{a,b} + \mathcal{O}(\|\Delta O\|_2^2), \qquad a, b \in \{1, \ldots, m+1\}. \tag{81}$$

where $a, b \in \{1, \ldots, m+1\}$ index the coordinates in that eigenbasis.

This shows the symmetric structure of Löwdin. The sensitivity to near-collinearity depends on the denominator in (81), so the effect of the noise is distributed across the spectrum of the whole retained block.

For Gram–Schmidt Orthogonalization. Let $O = T^\top T$ and $T = \mathrm{chol}(O)$, the Cholesky factorization of O; this is equivalent to Fisher Gram–Schmidt applied to the columns of S, where T collects the resulting projection and normalization coefficients, so the corresponding reduced coefficients are encoded by the upper-triangular factor T. If o_{ab} and t_{ab} denote the entries of O and T, then the Cholesky/Gram–Schmidt recursion is

$$t_{kk}^2 = o_{kk} - \sum_{s<k} t_{sk}^2, \qquad t_{kj} = \frac{o_{kj} - \sum_{s<k} t_{sk} t_{sj}}{t_{kk}}, \qquad j > k. \tag{82}$$

Using the first-order Taylor expansion of these relations gives

$$\Delta t_{kk} = \frac{\Delta o_{kk} - 2\sum_{s<k} t_{sk} \Delta t_{sk}}{2 t_{kk}} + \mathcal{O}(\|\Delta O\|_2^2), \tag{83}$$

and

$$\Delta t_{kj} = \frac{\Delta o_{kj} - \sum_{s<k} (\Delta t_{sk} t_{sj} + t_{sk} \Delta t_{sj}) - t_{kj} \Delta t_{kk}}{t_{kk}} + \mathcal{O}(\|\Delta O\|_2^2), \qquad j > k. \tag{84}$$

This shows the sequential noise structure of Gram–Schmidt: the noise at stage k is divided by the current pivot t_{kk}, and later quantities inherit earlier noise through the recursion.

Comparison in update. Fundamentally, both orthogonalization methods suffer under severe collinearity. LOCU addresses this via pairwise screening and compression, $|\varrho_{ij}| \geq 1 - \varepsilon_{\mathrm{dup}}$. However, this cannot fully prevent span-level near-collinearity, where a retained vector can align with the span of others much more tightly than the threshold $\varepsilon_{\mathrm{dup}}$.

In worst-case scenarios, the Gram matrix O becomes near-singular, destabilizing both methods. For Löwdin, this forces the minimum eigenvalue $\lambda_{\min}(O) \approx 0$, driving the denominator in (81) to approach zero. For Gram–Schmidt, the instability manifests in the sequential pivots. Rewriting (82) in Schur-complement form yields

$$t_{kk}^2 = o_{kk} - O_{1:k-1,k}^\top O_{1:k-1,1:k-1}^{-1} O_{1:k-1,k} = o_{kk}\big(1 - \rho_{k|1:k-1}^2\big). \tag{85}$$

Thus, span-level near-collinearity, $\rho_{k|1:k-1}^2 \approx 1$, drives the normalized residual t_{kk}^2/o_{kk} to near zero.

In such cases, both methods require regularization/damping to avoid division by near zero. However, their noise and bias propagation differ fundamentally as shown earlier. In Gram–Schmidt, the sequential structure causes both stochastic noise and the damping bias introduced at step k to be inherited and compounded by all subsequent steps $j > k$, leading to cumulative distortion in the resulting basis geometry. Conversely, Löwdin applies symmetric matrix-level regularization, e.g., $O + \varepsilon I$, which distributes both perturbation and damping effects across the entire eigenbasis without order-dependent accumulation. By treating all vectors simultaneously, Löwdin preserves the mutual symmetry and geometric integrity of the basis far more effectively than the inherently hierarchical Gram–Schmidt approach.

C.2.2 Empirical comparison

We evaluated LOCU on the BC-v3 Redun : 3 task by replacing Löwdin orthogonalization with Gram–Schmidt orthogonalization under three gradient orderings (original, random, and reverse), while injecting observation noise into each training batch following [16].

As shown in Figure 8, Löwdin orthogonalization generally performs below Gram–Schmidt index but above random and reverse, attributed to its symmetric distribution of perturbation-induced errors across gradient directions. In this task, the last three constraints in the indexed order are the loosest low-velocity constraints, and the differences among the Gram–Schmidt variants reflect their inherent ordering dependence. Under the indexed order, accumulated errors primarily propagate to these loose constraints, whereas under the reverse order, they affect the reward gradient and the tighter high-velocity constraints. The noise implementation of [16] adds perturbations of a fixed magnitude without accounting for the relative scale of each observation. At the highest noise level of 0.25, all methods degrade substantially, while the pronounced failure of Gram–Schmidt-random suggests that Gram–Schmidt orthogonalization can be more vulnerable than Löwdin when large stochastic perturbations distort the gradient geometry.

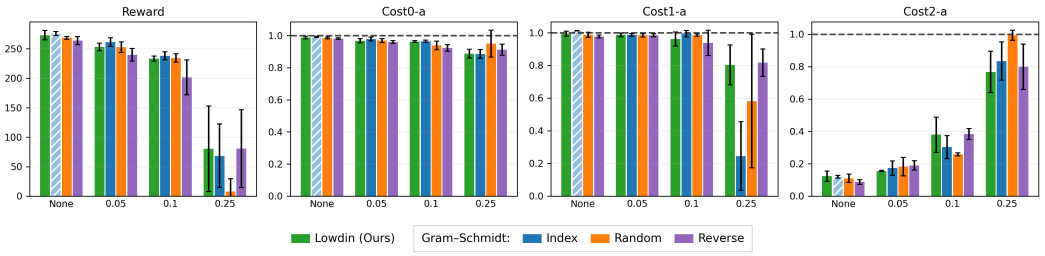


Figure 8: Performance comparison of Löwdin and Gram–Schmidt orthogonalization under observation noise on BC-v3 Redun 3.

C.3 Computational Complexity

C.3.1 Wall clock time results

Table 4 reports wall clock training time on BC-v2 and redun 3 variants of BC-v3, measured on a single machine with an NVIDIA RTX 4070 GPU, an Intel i7-13700K CPU, and 64GB RAM. PPOL-V and PPOL-GS are excluded from the redundancy results in the main text due to unstable learning in earlier stages, but we additionally trained them solely to compare wall clock time and include their times in the table. Wall clock time is logged with Weights and Biases [5].

Table 4: Wall-clock time on the main circle task.

Task	LOCU	SDAC(A)	PCRPO	CRPO	TRPOL -GS	TRPOL -V	PPOL -V	PPOL -GS
BC v2	17:08	18:20	17:30	17:23	16:08	16:27	17:11	18:29
BC v3 Redun 3	25:02	25:40	23:00	22:53	17:02	17:05	22:28	23:40

Table 5: Wall-clock time on Circle tasks with parallel processing.

Task	Redun	LOCU	SDAC(A)	CRPO	TRPOL	PPOL (bs=64)	PPOL (bs=128)
PointCircle	2	10:19	10:55	10:18	10:22	21:08	15:37
	3	10:31	11:08	10:19	10:25	21:13	15:40
	5	10:44	11:19	10:22	10:29	21:20	15:45

In the implementation used for the main-text experiments, LOCU computes all constraint natural gradients through CG and then solves the reduced program. Basis compression preserves the rows for all retained constraints, so the active-set linear system has dimension $|\mathcal{J}| \leq |\mathcal{K}|$, with a dense-solve worst-case term $\mathcal{O}(T_{\text{active}}|\mathcal{K}|^3)$. CG and the recovery simplex QP add separate costs, so the total cost can grow with the number of constraints. However, this overhead is not always dominant. In BC-v2, where only a small number of constraints are used, PPO-style baselines often spend more time on repeated mini-batch actor optimization and multiple constraint-critic updates, which can outweigh CG solves.

In BC-v3 Redun 3, where the number of constraints increases to nine, wall-clock time increases for most methods, with the largest increases observed for LOCU and SDAC(A). This reflects the cumulative cost of computing multiple constraint natural-gradient directions before the constrained update. The comparison should also be interpreted with implementation details: SDAC(A) uses OSQP and SciPy backends for its QP [22, 25], whereas LOCU is implemented in Python.

C.3.2 Parallel processing

To examine whether this bottleneck can be reduced, we construct a MuJoCo-JAX (MJX) [24] version of the CircleRun-v3 Redun task and implement LOCU, SDAC(A), CRPO, PPOLag, and

TRPOLag with vectorized constraint processing and end-to-end XLA compilation. This targets a common limitation of Python-based safe RL implementations, where most computations are executed sequentially and therefore use only part of the available hardware throughput. Table 5 reports the wall-clock time of the MJX implementation on PointCircle with redundancy levels 2, 3, and 5. The results show that full-batch second-order actor updates can be faster than minibatch-based first-order updates with parallel processing. LOCU, SDAC(A), CRPO, and TRPOL complete each run in roughly 10–11 minutes, whereas PPOLag takes 15–21 minutes depending on the minibatch size, denoted as bs in the table.

As expected, LOCU and SDAC(A), which compute natural-gradient directions for all constraints, show a larger redundancy-dependent increase than CRPO and TRPOL, which only compute single natural-gradient. Under the MJX implementation, however, this increase is small: from redundancy level 2 to 5, LOCU increases from 10:19 to 10:44 and SDAC(A) from 10:55 to 11:19, i.e., less than 30 seconds while the number of constraints grows from 6 to 15. By contrast, in the Python-based implementation in Table 4, increasing from BC-v2 with two constraints to BC-v3 Redun 3 with nine constraints increases LOCU and SDAC(A) by roughly 7–8 minutes. Compared with the substantially larger cost of PPOLag, the remaining difference between LOCU and the single-natural-gradient methods can be considered minor.

Overall, the results indicate that parallelized constraint processing substantially reduces wall-clock sensitivity to the number of constraints compared with sequential Python-based implementations. The same implementation strategy may therefore be useful for other multi-constraint safe RL methods or multi-objective settings. More broadly, the results weaken the view that second-order safe RL methods are impractical solely due to computational burden. When multiple constraints must be handled and constraint satisfaction is central to the task, LOCU or other trust-region-based second-order methods can remain practical options under parallel processing.

C.3.3 Inverse fisher preconditioner

Natural gradients are typically computed using conjugate gradient (CG), which needs to be applied separately to each gradient, the computational cost increases linearly with the number of constraints. An alternative option can be using a shared approximate Fisher inverse or preconditioner across all gradients. Since the FIM is defined solely by the outer product of policy score functions, independent of advantage estimates, layer-wise curvature approximations such as K-FAC and its variants can provide a single approximate inverse applicable to all reward and constraint gradients. Although these approaches can be more computationally efficient than using CG iterations depending on the network architecture and the number of constraints, further investigation is required to assess its approximation accuracy, numerical stability, and overall effectiveness in multi-constraint policy optimization.

C.4 Additional Experiments

We conduct additional experiments to evaluate LOCU from three perspectives. First, we examine the contribution of its algorithmic components through ablation studies. Second, we investigate hyperparameter sensitivity by varying the clustering threshold and KL limit. Finally, we evaluate robustness to redundant constraints by increasing the number of constraints while preserving the feasible region.

C.4.1 Component Analysis

We compare LOCU against two ablated variants and a CPO baseline. The two ablated variants in Figure 9 are labeled as `w/o Lowdin` and `w/o Lowdin, Screening & compression`. `w/o Lowdin` removes Löwdin orthogonalization and only normalizes each direction by its Fisher norm before applying the same reduced update. `w/o Lowdin, Screening & compression` further removes screening and clustering and uses all normalized constraint directions in the same update procedure. For the CPO baseline, we follow the original implementation where infeasibility and degenerate cases use a recovery direction formed by summing the gradients of currently violated constraints, and the resulting QP is solved with OSQP [22]. Removing Löwdin slows convergence relative to CPO and only recovers return late in training. When both Löwdin and screening with compression are removed, the early stage becomes highly seed dependent under simultaneous violations, and some runs finish with less than half of LOCU's final return. This is consistent with

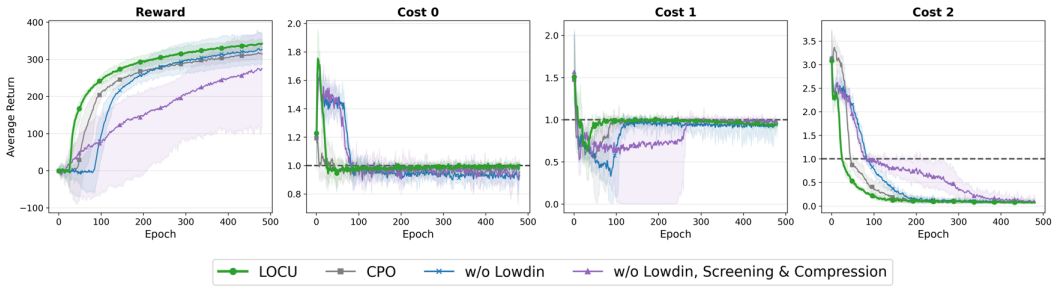


Figure 9: Learning curves for `w/o Lowdin` and `w/o Lowdin, Screening & compression`, compared with the CPO baseline.

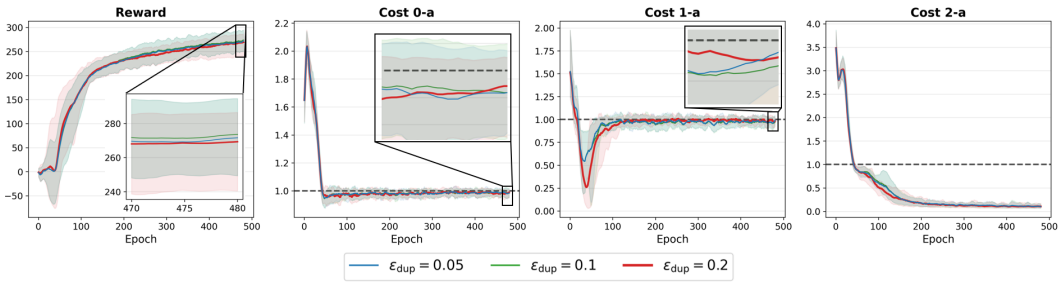


Figure 10: Ablation on the clustering threshold ε_{dup} for near-collinear constraint grouping on BC-v3 Redun : 3.

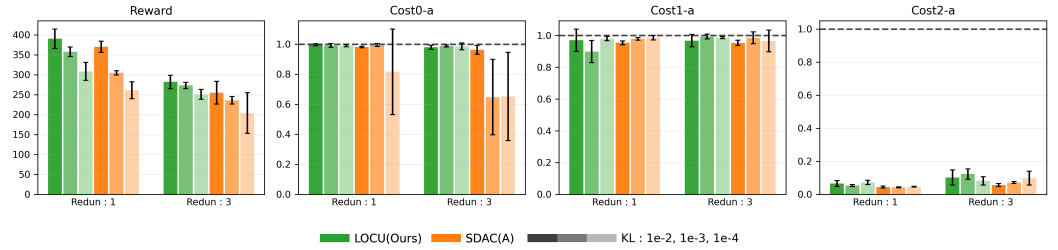


Figure 11: Sensitivity analysis on the KL divergence limit compared against SDAC(A) on BC-v3.

redundant and correlated constraint directions producing ill conditioned reduced systems and noisy recovery when they are treated as distinct correction directions without orthogonalization or reduction.

C.4.2 Hyperparameter Sensitivity

We test the cosine similarity threshold ε_{dup} that controls near-collinear clustering in Section 4.2. Since ε_{dup} is used to decide whether constraint gradients are treated as redundant, we evaluate it on BC-v3 Redun : 3, where redundancy is strongest. Figure 10 compares threshold values 0.05, 0.1, and 0.2. Differences appear mainly in the early recovery stage, while final returns remain very similar across all tested values. These gaps are small relative to typical across-seed variation and may narrow with additional seeds. Normalized costs converge below the limits in all cases, indicating that LOCU is not particularly sensitive to the exact choice of ε_{dup} within this range and does not require careful angle tuning to avoid ill-conditioned updates.

Next, we evaluate sensitivity to the trust region radius against SDAC(A) on the BC-v3 task. As the limit size decreases, a smaller KL limit generally reduces approximation error but yields less policy improvement per update. Since our task starts from an infeasible state, a smaller radius requires more updates to reach the feasible region, leaving fewer remaining updates for reward optimization. Both algorithms exhibit similar performance degradation in reward, while showing no further sensitivity in cost satisfaction, as shown in Figure 11.

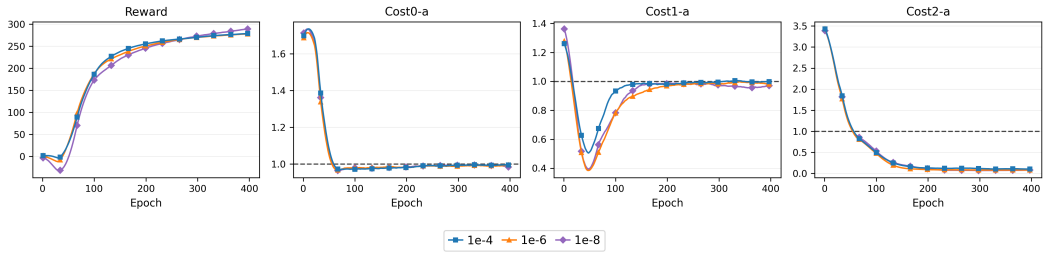


Figure 12: Ablation on threshold, damping, and CG tolerance values on BC-v3 Redun : 3.

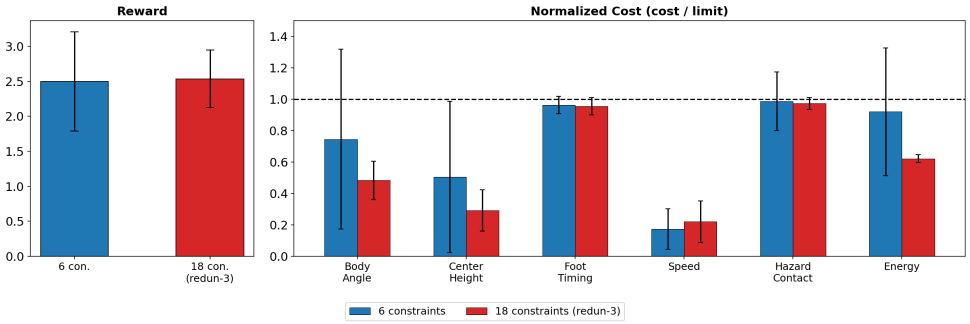


Figure 13: Final evaluation comparison between the original 6-constraint Cassie task and the redundant 18-constraint version.

Finally, we examine numerical hyperparameter sensitivity on the BC-v3 Redun : 3 task by simultaneously assigning 10^{-4} and 10^{-6} to the near-zero threshold, Fisher damping, and CG tolerance. The baseline setup uses default values 10^{-2} for the near-zero threshold, 10^{-1} for Fisher damping, and 10^{-8} for CG tolerance. As illustrated in Figure 12, smaller values increase reward slightly, while causing larger variance across runs. This behavior may attributed to Fisher damping, which produces a closer approximation to the true natural gradient while increasing sensitivity to near-zero eigenvalues of the Fisher matrix, making individual updates more accurate yet overall training less stable.

C.4.3 Robustness to Redundant Constraints

To evaluate robustness to a larger number of constraints without changing the feasible region, we introduce redundant constraint variants to the goal-navigation Cassie task. For each base constraint, we keep the original tight threshold and add two looser variants, increasing the total number of constraints from 6 to 18. For example, we use thresholds of $15°/20°/25°$ for body tilt, $13/15/18\,\mathrm{W}$ for energy, and $0.0/0.05/0.1\,\mathrm{m}$ for the hazard contact margin. This creates a setting with substantial redundancy and heterogeneity across constraint gradients.

Figure 13 compares the final evaluation results between the original 6-constraint Cassie task and the redundant 18-constraint version. LOCU preserves reward under $3\times$ constraint scaling while satisfying all constraints at convergence. Figure 20 further shows that the learning dynamics remain stable after adding redundant constraints, indicating that the additional constraints do not substantially degrade optimization.

Table 6 reports the screening statistics during training in the 18-constraint setting. Even with 18 nominal constraints, only a small subset is active at each update, and redundancy compression further reduces the basis dimension. By the end of training, the average active set size is 4.9 and the average basis dimension is 2.8, so the practical optimization remains compact.

C.5 Learning curves

We report learning curves for the experiments in the main text and Appendix. Costs are normalized by their corresponding limits, so values above 1 indicate constraint violations.

Table 6: Screening statistics during training in the 18-constraint redundant Cassie task.

Phase	Basis dim.	Active set	Redun.	Inactive	Zero-grad
0–20%	5.7 ± 0.9	9.5 ± 1.6	3.8 ± 1.0	8.5 ± 1.6	2.5 ± 1.2
20–40%	3.8 ± 1.5	6.5 ± 3.1	2.7 ± 1.6	11.5 ± 3.1	3.2 ± 1.7
40–60%	3.9 ± 1.7	6.6 ± 4.0	2.7 ± 1.8	11.4 ± 4.0	1.6 ± 1.1
60–80%	3.1 ± 1.5	5.5 ± 4.2	2.4 ± 1.7	12.5 ± 4.2	1.2 ± 0.5
80–100%	2.8 ± 1.4	4.9 ± 4.0	2.1 ± 1.5	13.1 ± 4.0	0.9 ± 0.4

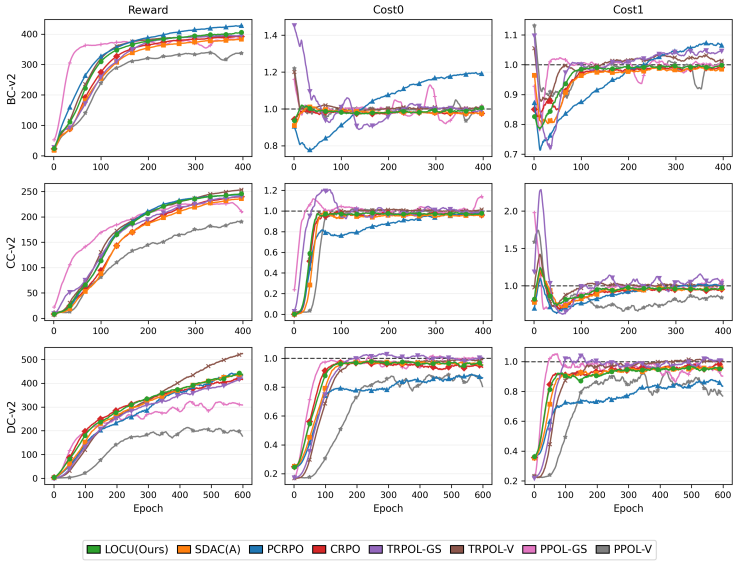


Figure 14: Learning curves on the Bullet Safety Gym v2 suite for BC, CC, and DC.

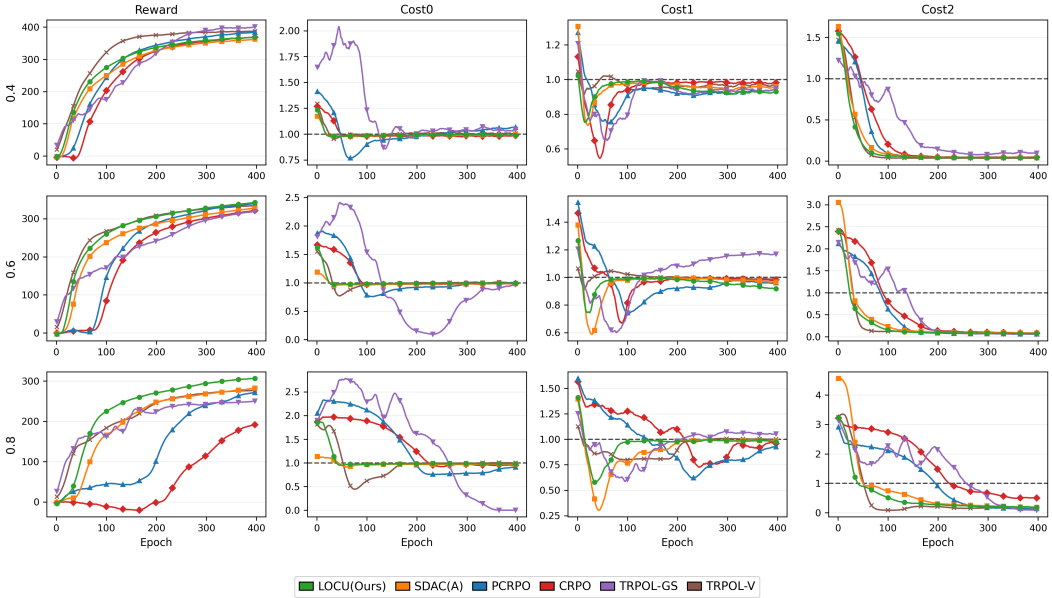


Figure 15: Learning curves on BC-v3 under the low velocity sweep from the v2 case to $\kappa_{\text{lo}} \in \{0.4, 0.6, 0.8\}$.

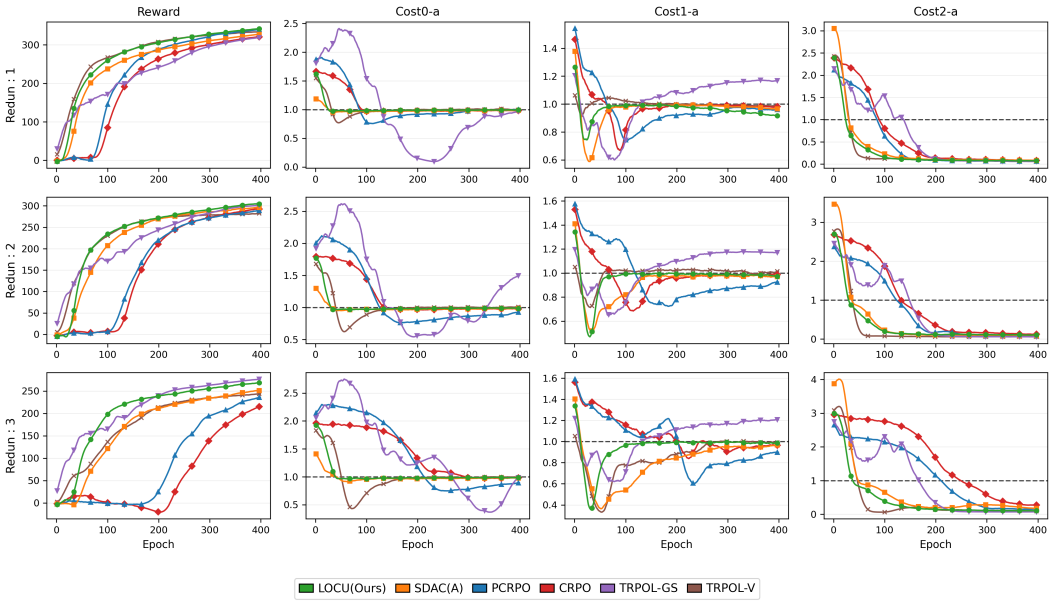


Figure 16: Learning curves on BC-v3 under redundancy variants redun : 1, redun : 2, and redun : 3. The "-a" denotes the tightest redundant constraint.

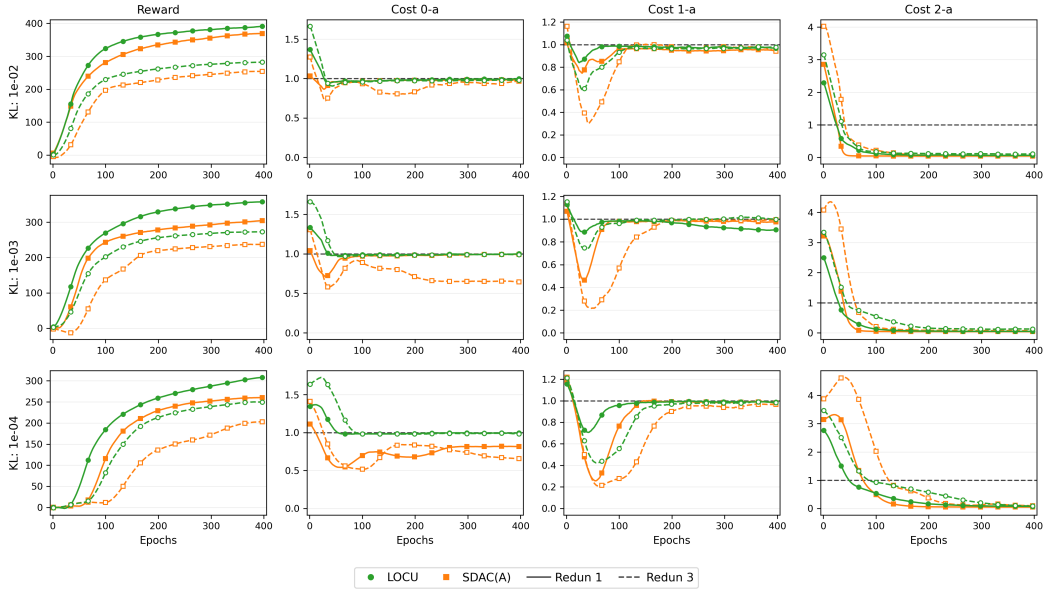


Figure 17: Learning curves of LOCU and SDAC(A) under different KL limits on BC-v3 under redundancy variants redun : 1 and redun : 3.

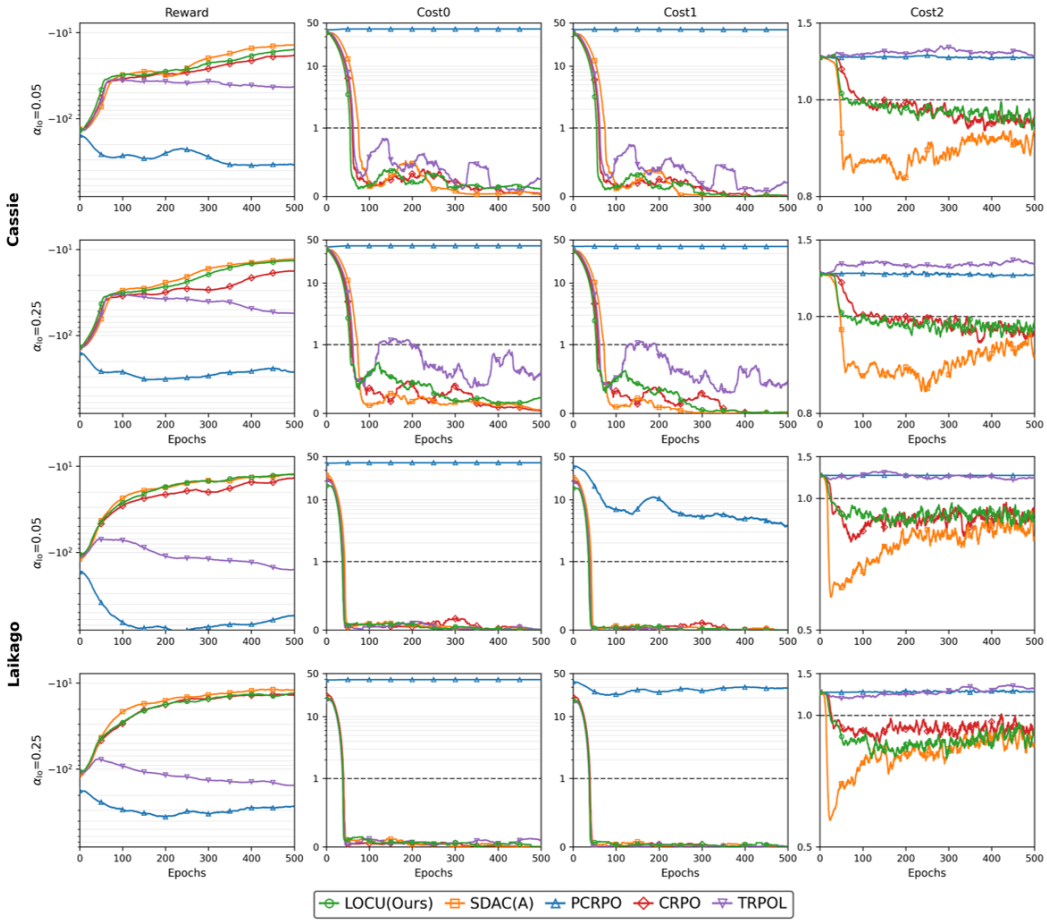


Figure 18: Learning curves on the locomotion tasks with Cassie and Laikago under three simultaneous constraints.

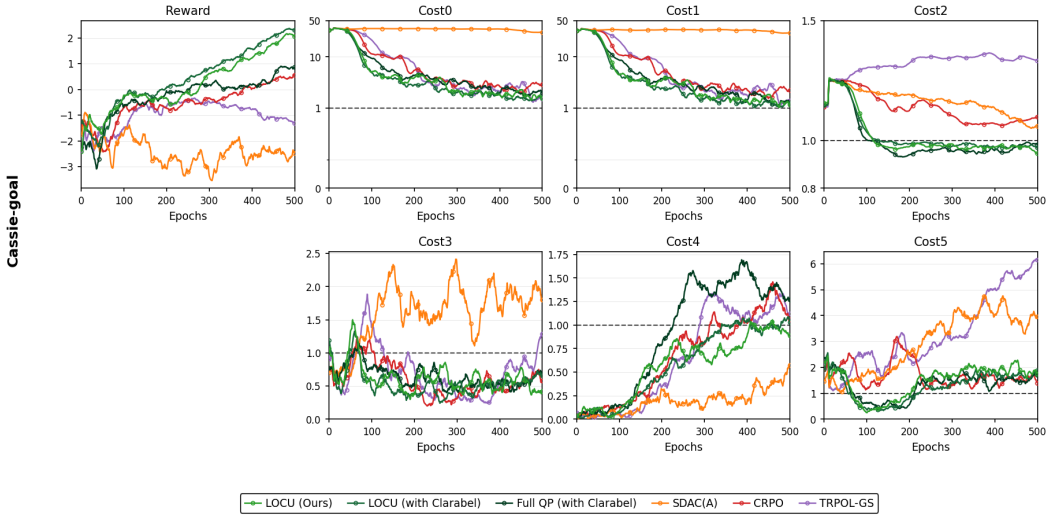


Figure 19: Learning curves on the goal-navigation task with Cassie under additional three simultaneous constraints.

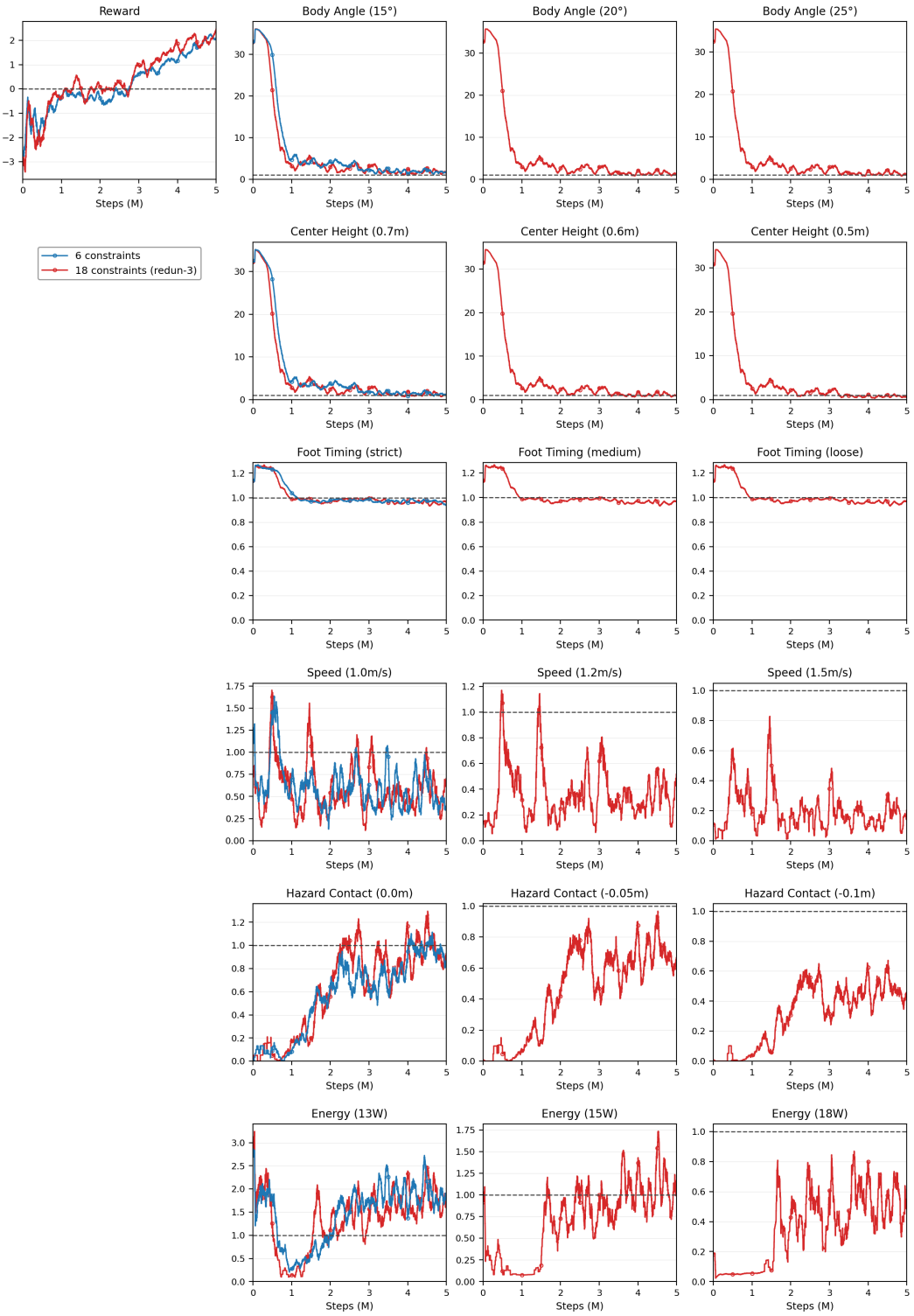


Figure 20: Learning curves for the goal-navigation task with Cassie of 18-constraint version.

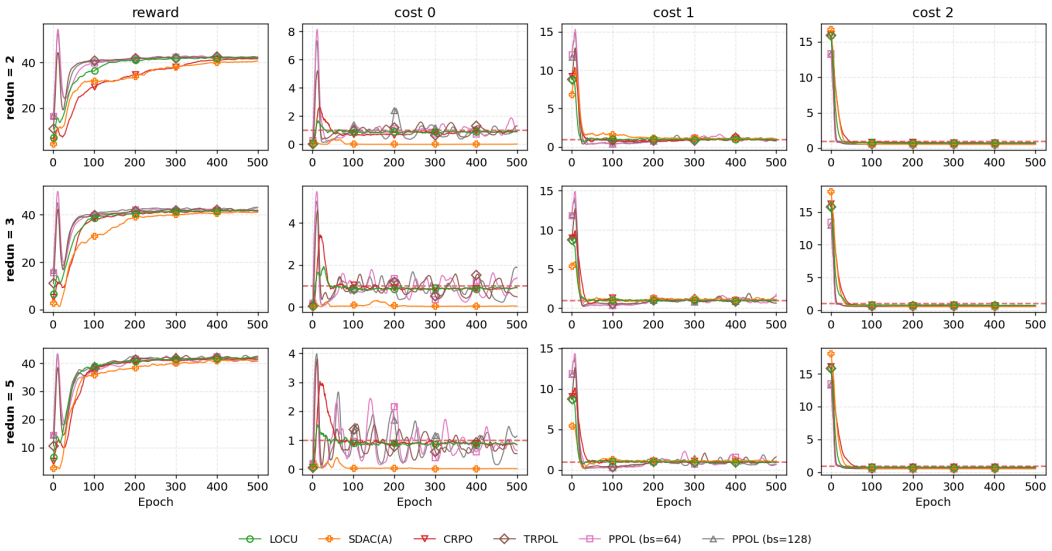


Figure 21: Learning curves for the MJX implementation of circle tasks.

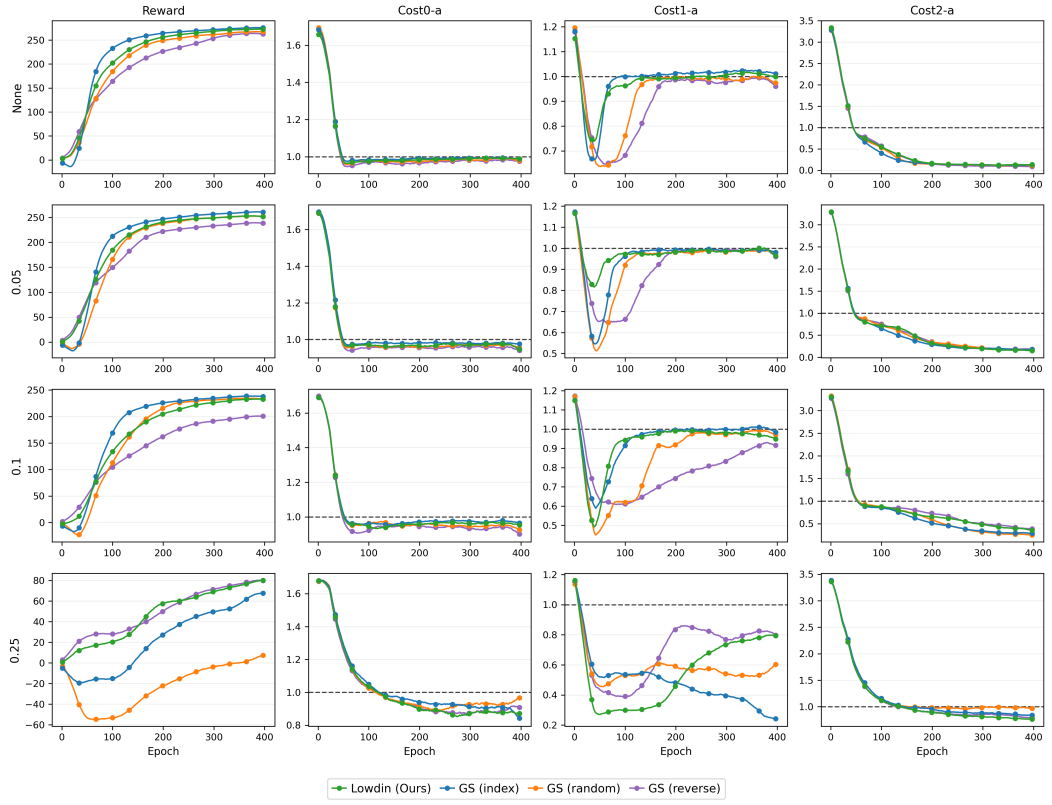


Figure 22: Learning curves for the orthogonalization comparison.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes],

   Justification: The abstract and Section 1 state the proposed LOCU framework, including Fisher-orthonormal reduced coordinates, constraint screening and compression, feasibility recovery, and the empirical scope over multi-constraint benchmarks.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The Limitations and future works section discusses local feasibility, lack of global convergence guarantees, and computational cost.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: The main text states the assumptions and theoretical results, and Appendix A.2 provides the proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Section 5 and Appendix B provide the tasks, constraints, baselines, evaluation protocol, implementation details, and hyperparameters.

   Guidelines:

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

5. **Open access to data and code**

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

Answer: [Yes]

The code is publicly available in the GitHub repository linked in the abstract, along with setup and reproduction instructions provided in the respective experiment directories.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Table 2 and Appendix B provide the model architectures, hyperparameters, task settings, and evaluation procedures.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Results are averaged over five training seeds and 100 evaluation episodes with 10 rollout seeds, and error bars indicate standard deviation.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Appendix C.3.1 reports the hardware configuration and wall-clock measurements.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The work studies safe policy optimization algorithms in simulated control benchmarks and does not involve human-subject data, personally identifiable information, or high-risk released models or datasets.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: The work may benefit safe control applications by improving policy optimization under multiple constraints, but its contributions are methodological, with no direct societal impacts identified.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release high-risk data or models such as pretrained language models, image generators, scraped datasets, or dual-use artifacts.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The original works are cited in the paper, and the GitHub repository provides attribution and license information for reused code.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: The GitHub repository linked in the abstract provides instructions for the released code in each corresponding directory.

    Guidelines:

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

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

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

    Answer:[N/A]

    Justification: The paper does not involve crowdsourcing or human-subject research.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve human-subject research, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: We only used LLMs for editing and formatting.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.