DO WE NEED COST CRITICS? LOCAL COST RELABELING FOR OFFLINE SAFE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline safe reinforcement learning commonly relies on learned cost-to-go functions, feasibility models, or adaptive penalty mechanisms to enforce cumulative-cost constraints. We ask how much of this machinery is necessary when safety violations are directly observed in the offline data. We study a simple alternative, RELABEL(COST), that retains the full dataset, replaces the reward on positive-cost transitions by a fixed negative value derived from the reward scale, and then trains a behavior-regularized offline RL policy. The method uses no cost critic, no feasibility estimator, and no safety-specific penalty tuning. In backbone-matched continuous-control experiments, RELABEL(COST) consistently satisfies the safety budget and achieves near-zero cost on most tasks while remaining competitive with, and often outperforming, adaptive additive penalties and cost-critic-based relabeling. These comparisons also reveal an important role for strong behavior regularization, with local cost supervision identifying undesirable transitions while regularization limits policy improvement to regions supported by the offline data. Our analysis characterizes when fixed local penalties align with zero-cost policy optimization, explains how delayed cost events propagate through ordinary reward value learning, and shows that safe behavior can in principle be recovered even when complete trajectories in the dataset are infeasible. Together, the results suggest that direct local cost supervision combined with strong offline regularization can provide a simple and robust alternative to explicit cumulative-cost modeling in offline safe RL.

1 INTRODUCTION

Offline reinforcement learning (RL) asks a learner to improve a policy using only previously collected data, without further interaction with the environment (Levine et al., 2020). This is particularly appealing in safety-critical domains, where collecting new data may be costly or dangerous. In these settings, however, high return is not enough. The deployed policy must also satisfy explicit safety constraints, commonly modeled through constrained Markov decision processes (CMDPs) (Altman, 1999; García and Fernández, 2015). Offline safe RL therefore combines policy improvement under distribution shift with reliable control of cumulative safety costs.

A growing literature has addressed this problem from several directions. Early offline safe RL methods modify value learning or stationary-distribution optimization to enforce cost constraints, including CPQ and COptiDICE (Xu et al., 2022; Lee et al., 2022). Sequence-model approaches condition decisions on desired reward and cost returns (Liu et al., 2023), while more recent methods explicitly model feasible regions (Zheng et al., 2024), reshape or augment the offline data distribution (Yao et al., 2024), learn constraint-conditioned value functions (Liu et al., 2025), regularize policies with generative models (Guo et al., 2025), or adapt reward–cost tradeoffs through policy switching or online optimization (Chemingui et al., 2025b;a). These approaches reflect an important fact about the problem: safety is a trajectory-level property, and a policy can violate a budget because of consequences that unfold many steps after an individual decision.

At the same time, the offline dataset often contains safety information at a much finer resolution. In the DSRL benchmark, for example, each transition is accompanied by a cost signal, and the benchmark contains heterogeneous trajectories ranging from relatively safe to highly unsafe behavior (Liu et al., 2024). A trajectory that exceeds the deployment budget need not be useless. It may contain

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

long stretches of productive zero-cost behavior together with a small number of transitions on which cost is actually observed. Discarding the complete trajectory throws away both kinds of information. Conversely, learning a cumulative-cost model introduces another estimation problem whose predictions may subsequently control policy updates, feasibility tests, or penalty selection.

This motivates a deliberately simple question: *how far can we get by using the observed local cost signal directly?* We study RELABEL(COST), which retains every transition in the offline dataset, leaves the reward unchanged when the observed cost is zero, and replaces the reward by a fixed negative value when the observed cost is positive. We then train an otherwise standard behavior-regularized offline RL algorithm on the relabeled data. Our implementation uses ReBRAC (Tarasov et al., 2023a), a strong minimalist offline RL method built on the same behavior-regularization principle that underlies TD3+BC (Fujimoto and Gu, 2021). No cost critic, feasibility model, Lagrange multiplier, or deployment budget is used during training.

Reward penalties themselves are not new in safe or offline RL. CPQ modifies value learning to penalize unsafe or unsupported actions (Xu et al., 2022). Our focus is on a different design choice. Rather than first predicting cumulative future cost and then deciding which transitions should be penalized, RELABEL(COST) uses only the cost event that is directly observed in the data. Temporal credit assignment is not removed: after relabeling, the ordinary reward critic still propagates the consequences of a later cost event backward through Bellman learning. The safety signal itself, however, is grounded in observed transitions rather than in a second learned long-horizon critic.

Our experiments suggest that this distinction matters. In backbone-matched comparisons on continuous-control offline safe RL tasks, direct cost relabeling is consistently safe and competitive in reward with adaptive additive penalties and cost-critic relabeling, while requiring neither penalty search nor an additional critic. It is robust to changes in the penalty scale. More importantly, it continues to learn safe policies after all trajectories below increasingly stringent cumulative-cost thresholds are removed from the training data. These results suggest that complete feasible trajectories are not always necessary when useful local behavior and observed cost events remain in the dataset. Strong behavior regularization then plays a complementary role by limiting policy improvement toward unsupported actions, a central concern throughout offline RL (Levine et al., 2020; Kumar et al., 2020; Fujimoto and Gu, 2021; Kostrikov et al., 2022; Tarasov et al., 2023a).

We develop theory around this empirical phenomenon rather than treating the fixed penalty as a universal solution to constrained RL. First, we give a finite-penalty bound showing how approximate optimization of the relabeled objective controls the expected number of cost events and hence cumulative cost and budget-violation probability. Second, an exact event-price characterization identifies when the relabeled objective has the same optimum as strict zero-cost policy optimization. Third, a delayed-cost analysis shows how a penalty applied only when cost is observed can create value separation at earlier decisions through ordinary Bellman propagation. Finally, a constructive finite-data result shows that complete feasible episodes are not information-theoretically necessary for recovering safe behavior when the relevant transition-level structure is covered.

Our contributions are as follows:

- We introduce and systematically study a fixed local-cost reward transformation for offline safe RL. The method uses observed cost events and a behavior-regularized offline RL backbone, without learning a separate cost critic or tuning a safety-specific multiplier.
- We provide backbone-matched experiments separating the effects of reward replacement, additive penalties, cost-critic relabeling, penalty scale, and the availability of feasible trajectories.
- We develop a theoretical account of when local cost supervision is sufficient, including finite-penalty safety bounds, an exact event-price characterization, delayed-cost propagation, and show how safe policies can be recovered from transition evidence embedded in infeasible episodes.

2 PROBLEM SETUP

We consider a finite-horizon constrained Markov decision process (Altman, 1999) with horizon H, observed time or phase, state space $\mathcal{S} \subseteq \mathbb{R}^{d_s}$, action space $\mathcal{A} \subseteq \mathbb{R}^{d_a}$, initial-state distribution ξ, and transition, reward, and cost kernels. A trajectory generated by a policy π is $\tau = (s_0, a_0, r_0, c_0, \ldots, s_{H-1}, a_{H-1}, r_{H-1}, c_{H-1}, s_H)$. Throughout the main analysis we use finite-

horizon undiscounted returns and assume

$$r_- \le r_t \le r_+, \qquad 0 \le c_t \le c_{\max}, \qquad c_{\max} > 0, \qquad \Delta_r = r_+ - r_-. \tag{1}$$

If an episode terminates before H, it may be padded with absorbing zero-reward, zero-cost transitions. A positive cost event does not itself terminate the process. For a policy π, define the expected reward and cumulative cost

$$J_R^\pi = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{H-1} r_t\right], \qquad C_\tau = \sum_{t=0}^{H-1} c_t, \qquad J_C^\pi = \mathbb{E}_{\tau\sim\pi}[C_\tau]. \tag{2}$$

The standard budget-constrained objective is $J_B^\star = \sup_{\pi\in\Pi:\, J_C^\pi \le B} J_R^\pi$, where $B \ge 0$ is the *deployment budget* and Π is the policy class. We call a policy budget-feasible when $J_C^\pi \le B$, and a recorded trajectory budget-feasible when $C_\tau \le B$. We also write $p_B^\pi = \mathbb{P}_{\tau\sim\pi}(C_\tau > B)$ for the *probability of exceeding the budget on a trajectory*.

Our method is built around the observed occurrence of positive-cost transitions. Define

$$v_t = \mathbf{1}\{c_t > 0\}, \qquad K_\tau = \sum_{t=0}^{H-1} v_t, \qquad n^\pi = \mathbb{E}_{\tau\sim\pi}[K_\tau]. \tag{3}$$

We will use the strict-safety subclass $\Pi_0 = \{\pi \in \Pi : n^\pi = 0\}$ and assume $\Pi_0 \neq \varnothing$. Its optimal return is $J_0^\star = \sup_{\pi\in\Pi_0} J_R^\pi$. Because costs are nonnegative, every policy in Π_0 has zero cumulative cost almost surely and is feasible for every $B \ge 0$. Thus $J_0^\star \le J_B^\star$. We do not assume that these two quantities are equal. The gap $J_B^\star - J_0^\star$ is the *price of targeting strict zero-cost behavior* rather than optimally spending the available budget. A central empirical question in this work is whether this price is small on the benchmark tasks we study.

The learner receives a fixed offline dataset $\mathcal{D} = \{(s_i, a_i, r_i, c_i, s_i', d_i, t_i)\}_{i=1}^N$ collected before training by one or more unknown behavior policies. Here d_i records termination information and t_i is the observed phase. We make no assumption that $\mathcal{D}$ contains a budget-feasible complete trajectory. This is stronger than the usual offline-RL difficulty of distribution shift (Levine et al., 2020): the target policy may need to recombine locally useful behavior from trajectories that are globally infeasible. Unless stated otherwise, policies are Markov in state and observed time.

The following elementary relations connect the local event count to the cumulative-cost and trajectory-level safety criteria used later.

Lemma 1 (Events, cost, and budget failure)**.** *Let* $k_B = \left\lfloor \frac{B}{c_{\max}} \right\rfloor + 1$. *Then, for every policy* π,

$$J_C^\pi = 0 \iff n^\pi = 0 \iff C_\tau = 0 \text{ almost surely}, \tag{4}$$

and $J_C^\pi \le c_{\max} n^\pi$, $p_B^\pi \le \min\left\{1, \frac{n^\pi}{k_B}\right\}$.

The first inequality shows that controlling the expected number of positive-cost events controls expected cumulative cost. The second converts event-count control into a distribution-free bound on budget-violation probability. Our theoretical analysis will therefore study how the relabeled objective controls n^π, while the experiments report the original reward and cost criteria used by the offline safe RL benchmark (Liu et al., 2024).

3 Local Cost Relabeling

Our method uses the cost labels already present in the offline dataset. Rather than learning a separate critic to estimate cumulative future cost, we modify the reward only on transitions for which a positive cost is directly observed, and then apply a standard behavior-regularized offline RL algorithm to the resulting dataset. Let

$$P_{\mathcal{D}} = \max_{i\in\mathcal{D}} |r_i|, \qquad \widetilde{r}_i = \begin{cases} r_i, & c_i = 0, \\ -P_{\mathcal{D}}, & c_i > 0. \end{cases} \tag{5}$$

Thus every zero-cost transition keeps its original reward, while every positive-cost transition receives the same negative reward. The scale $P_{\mathcal{D}}$ is computed once from the dataset and requires no safety-specific hyperparameter search. If all recorded rewards are zero, we use a fixed positive fallback value $P_{\min}$. This transformation is intentionally local. It does not label an entire trajectory as unsafe because the trajectory eventually violates a cost budget, nor does it use a learned estimate of future cost to decide which transitions should be penalized. A trajectory that contains a cost event may still contain many useful zero-cost transitions, and these transitions retain their original rewards.

3.1 DIRECT SUPERVISION FROM OBSERVED COST EVENTS

For the analysis, it is useful to replace $P_{\mathcal{D}}$ by a generic penalty $P \geq 0$. Writing $v_t = \mathbf{1}\{c_t > 0\}$, the transformed reward is $\widetilde{r}_{P,t} = r_t(1 - v_t) - Pv_t$. If $A^\pi = \mathbb{E}_\pi[\sum_t r_t(1 - v_t)]$ denotes the reward collected on zero-cost transitions and $n^\pi = \mathbb{E}_\pi[\sum_t v_t]$ is the expected number of positive-cost events, then the corresponding population objective is simply $J_P^\pi = A^\pi - Pn^\pi$. This form makes clear what the transformation changes. A policy that never incurs a positive cost keeps its original return, while policies that encounter cost events are charged P for each such event. In the implementation, $P = P_{\mathcal{D}}$. We use the generic P only to study how the strength of the penalty affects the learned objective. Although the supervision is local, the effect of the penalty is not limited to the transition on which the cost is observed. The ordinary reward critic still performs temporal credit assignment. For a fixed policy π, its Bellman recursion is $Q_{P,t}^\pi(s,a) = \mathbb{E}\left[\widetilde{r}_{P,t} + V_{P,t+1}^\pi(s') \mid s, a\right]$. Consequently, an action with zero immediate cost can receive a low value if it reliably leads to a later penalized transition. Conversely, a state appearing in an unsafe recorded trajectory can still have high value if the dataset contains a better continuation from that state. RELABEL(COST) therefore avoids learning a separate cumulative-cost critic without giving up multi-step credit assignment.

3.2 BEHAVIOR-REGULARIZED POLICY IMPROVEMENT

Reward labeling alone does not address the main statistical difficulty of offline RL. A learned policy can exploit errors in the value function by selecting actions that are poorly represented in the dataset (Levine et al., 2020; Kumar et al., 2020). We therefore combine the transformed rewards with behavior-regularized policy improvement.

At an abstract level, the learner approximately optimizes $J_P^\pi - \beta\Omega(\pi)$, where $\Omega(\pi)$ measures deviation from the behavior represented in the offline data and $\beta \geq 0$ controls the strength of that regularization. The penalty P and the regularization parameter β play different roles. The former changes the relative value of transitions on which a cost event is observed; the latter limits how far policy improvement moves away from supported behavior.

In our experiments we use ReBRAC (Tarasov et al., 2023a), which applies behavior regularization to both the actor and critic. This choice is important empirically. The relabeled rewards provide a direct signal for avoiding observed cost events, while ReBRAC helps prevent the policy from replacing those actions with unsupported alternatives whose values are unreliable.

The complete procedure is simple. We compute $P_{\mathcal{D}}$, replace the reward by $-P_{\mathcal{D}}$ on every transition with $c_i > 0$, retain all transitions and their dynamics, and train one behavior-regularized offline RL policy on the transformed dataset. No cumulative-cost critic, feasibility model, Lagrange multiplier, or deployment budget is required during training.

4 WHY CAN LOCAL COST RELABELING WORK?

RELABEL(COST) modifies rewards only on transitions with observed positive cost, whereas safety depends on cumulative trajectory cost. How much safety can a finite replacement penalty provide? When does the relabeled objective recover the best zero-cost policy exactly? How can a penalty at a later cost event influence earlier decisions? Can local transition information remain useful when no complete safe trajectory is available?

We first study the population relabeled objective, separating reward replacement from the statistical problem of learning from a fixed dataset. We then connect the guarantees to temporal credit assignment, behavior regularization, and the information contained in infeasible trajectories.

4.1 FINITE PENALTIES CONTROL COST EVENTS

We begin with a guarantee that holds for any finite penalty P. Recall the relabeled objective $J_P^\pi = A^\pi - Pn^\pi$. Equivalently, $J_P^\pi = J_R^\pi - \mathbb{E}_{\tau\sim\pi}[\sum_{t=0}^{H-1}(P + r_t)v_t]$. Because $r_t \geq r_-$, each positive-cost event reduces the relabeled return by at least $P + r_-$. The result below compares the learned policy with any zero-cost policy available to the learner. The quantity ϵ_{RL} collects the error incurred in learning from the offline data, including optimization, estimation, and distribution-shift error.

Theorem 1 (Finite-penalty control)**.** *Assume $P + r_- > 0$, and let $\pi_0 \in \Pi_0$ be any zero-cost comparator. Suppose the policy $\widehat{\pi}$ returned by the offline learner satisfies $J_P^{\widehat{\pi}} \geq J_P^{\pi_0} - \epsilon_{\mathrm{RL}}$. Then $n^{\widehat{\pi}} \leq \frac{Hr_+ - J_R^{\pi_0} + \epsilon_{\mathrm{RL}}}{P + r_-}$. Consequently,*

$$J_C^{\widehat{\pi}} \leq \frac{c_{\max}\big(Hr_+ - J_R^{\pi_0} + \epsilon_{\mathrm{RL}}\big)}{P + r_-}, \qquad p_B^{\widehat{\pi}} \leq \min\left\{1,\ \frac{Hr_+ - J_R^{\pi_0} + \epsilon_{\mathrm{RL}}}{k_B(P + r_-)}\right\}, \tag{6}$$

where $k_B = \lfloor B/c_{\max} \rfloor + 1$ as in Lemma 1.

Theorem 1 gives a direct interpretation of the replacement penalty. If the offline learner approximately matches a zero-cost comparator under the relabeled objective, increasing P reduces the expected number of positive-cost events at rate $O(1/P)$. Lemma 1 then gives corresponding bounds on expected cumulative cost and budget-violation probability. Proof is in Appendix A.1.

The guarantee does not require the practical penalty $P_{\mathcal{D}}$ to exceed an unknown exact threshold, so it applies to the finite fixed penalties used in our experiments. The quality of the zero-cost comparator also matters. a high-reward comparator leaves less room for a policy to compensate for cost events through additional task reward. RELABEL(COST) does not in general solve the budget-constrained problem defining $J_B^\star$. It favors policies with few or no positive-cost events, with Equation 6 quantifying the resulting safety control relative to a zero-cost comparator. We next ask when the penalty is large enough that optimizing the relabeled objective recovers the optimal zero-cost policy exactly.

4.2 WHEN IS THE REPLACEMENT PENALTY EXACT?

Theorem 1 shows that any sufficiently well-optimized finite penalty controls the expected number of cost events. A stronger question is whether there is a penalty above which the relabeled problem becomes exactly equivalent to optimizing reward over zero-cost policies.

The answer depends on how much additional reward a policy can obtain by accepting positive-cost events. Recall that A^π denotes the expected reward collected on zero-cost transitions and that the relabeled objective is $J_P^\pi = A^\pi - Pn^\pi$. We define the *event price*

$$\rho_0 = \sup_{\pi \in \Pi:\, n^\pi > 0} \frac{[A^\pi - J_0^\star]_+}{n^\pi}, \tag{7}$$

with the supremum of an empty set defined as zero. The quantity ρ_0 measures the largest reward advantage over the optimal zero-cost return that can be obtained per expected cost event.

Theorem 2 (Exact penalty threshold)**.** *If $P > \rho_0$, then for every policy $\pi \in \Pi$,*

$$J_P^\pi \leq J_0^\star - (P - \rho_0)n^\pi. \tag{8}$$

Consequently, $\sup_{\pi \in \Pi} J_P^\pi = J_0^\star$, and every maximizer of J_P^π, whenever a maximizer exists, is zero-cost and attains original return $J_0^\star$. Conversely, if $0 \leq P < \rho_0$, there exists a policy π with $J_P^\pi > J_0^\star$. At $P = \rho_0$, positive-cost policies may tie with a zero-cost optimum.

Theorem 2 gives an exact characterization of the penalty scale needed for objective alignment. A positive-cost policy can outperform the zero-cost optimum only if the additional reward it obtains is large enough to compensate for the penalty paid on its cost events. Once $P > \rho_0$, every positive-cost policy is separated from the zero-cost optimum by the margin in Equation 8.

The event price is an analytical quantity rather than a parameter estimated by the algorithm. It depends on the policy class and the unknown environment, so Theorem 2 should not be interpreted as a prescription for choosing $P_{\mathcal{D}}$. Theorems 1 and 2 describe complementary regimes. The finite-penalty result applies below the exact threshold as well, while the event-price result identifies when positive-cost behavior cannot improve the relabeled objective at all. The penalty-scaling experiments in Section 5 vary the penalty magnitude but do not estimate ρ_0 or identify its exact threshold.

How costly is strict zero-cost optimization? The zero-cost objective can be more conservative than the original budgeted problem. The event price also gives a simple way to quantify this gap when every positive cost event has a minimum magnitude. Suppose there exists $c_{\min} > 0$ such that $c_t > 0$ implies $c_t \geq c_{\min}$, and write $r_+^+ = \max\{r_+, 0\}$.

Proposition 1 (Budget value gap)**.** *For every budget $B \geq 0$, $0 \leq J_B^\star - J_0^\star \leq \frac{B}{c_{\min}}\big(\rho_0 + r_+^+\big)$. In particular, for binary unit costs $c_t \in \{0, 1\}$, $J_B^\star - J_0^\star \leq B\big(\rho_0 + r_+^+\big)$.*

The bound separates two ways in which allowing positive cost can improve reward. The term ρ_0 measures how much additional reward can be collected on zero-cost transitions per expected cost event, while r_+^+ bounds the reward collected on the cost-bearing transitions themselves. Thus strict zero-cost optimization loses little relative to the budgeted optimum when the available budget is small or when positive-cost behavior offers little reward advantage per event. It is generally conservative and does not imply that $J_B^\star - J_0^\star$ is small on a particular benchmark.

4.3 How does a local penalty affect earlier decisions?

RELABEL(COST) changes the reward only when a positive cost is observed. This may appear too local when the decision responsible for a violation occurs several steps earlier. The key point is that the relabeling is local, but the value function is not. Bellman backups propagate a later negative reward to earlier actions whenever those actions reliably lead to the penalized event.

We make this mechanism explicit in a simple delayed-cost model. Consider a deterministic segment of length $L \geq 2$, with dynamics $x_{t+1} = f_t(x_t, a_t)$. Costs are zero until the final transition, where a positive event occurs if the terminal state lies outside a safe set G:

$$c_t = 0 \quad \text{for } t < L-1, \qquad c_{L-1} = c_{\mathrm{e}} \mathbf{1}\{x_L \notin G\}, \qquad c_{\mathrm{e}} > 0. \tag{9}$$

Let $\mathcal{C} \subseteq \mathcal{A}$ denote the covered action set, representing actions supported by the offline data. Define the states from which the terminal event can still be avoided using covered actions by

$$\mathcal{K}_L = G, \qquad \mathcal{K}_t = \{x : \exists a \in \mathcal{C} \text{ such that } f_t(x,a) \in \mathcal{K}_{t+1}\}. \tag{10}$$

Thus $x \in \mathcal{K}_t$ means that at least one covered continuation can still complete the segment without a cost event. Let $Q_t^{\mathcal{C}}$ and $V_t^{\mathcal{C}}$ be the optimal covered values under the relabeled reward. They satisfy

$$Q_t^{\mathcal{C}}(x,a) = \widetilde{r}_{P,t}(x,a) + V_{t+1}^{\mathcal{C}}(f_t(x,a)), \qquad V_t^{\mathcal{C}}(x) = \max_{a \in \mathcal{C}} Q_t^{\mathcal{C}}(x,a), \qquad V_L^{\mathcal{C}} = 0. \tag{11}$$

The following result quantifies the value separation created by a delayed cost event.

Lemma 2 (Delayed-event separation)**.** *Let $x \in \mathcal{K}_t$ and write $T = L - t$. Suppose a covered action a_s satisfies $f_t(x,a_s) \in \mathcal{K}_{t+1}$, while another covered action a_b satisfies $f_t(x,a_b) \notin \mathcal{K}_{t+1}$. Then $Q_t^{\mathcal{C}}(x,a_s) \geq T r_-$, $Q_t^{\mathcal{C}}(x,a_b) \leq (T-1)r_+ - P$, and therefore $Q_t^{\mathcal{C}}(x,a_s) - Q_t^{\mathcal{C}}(x,a_b) \geq G_P$, and $G_P := P + r_- - (L-1)\Delta_r$.*

Lemma 2 isolates the temporal-credit-assignment mechanism. A covered action that preserves the possibility of avoiding the event can obtain at least the minimum reward for each of the remaining T steps. Once an action leaves $\mathcal{K}_{t+1}$, no covered continuation can avoid the terminal cost event. When $G_P > 0$, the later cost event creates a strict value preference for the safe action even though both actions may have zero immediate cost. This distinction is important for interpreting RELABEL(COST). The method does not assume that the transition receiving the penalty is the decision that caused the violation. It requires only that the ordinary reward critic can propagate the relabeled outcome backward through supported transitions. In this sense, the reward critic performs the temporal prediction, while the observed cost determines where the safety supervision enters.

The same idea yields a global margin when such delayed-cost segments repeat. Suppose an episode contains J identical segments of length L, and the environment resets to the same initial state after each segment. Let $v_0^\star$ denote the optimal zero-cost original return in one segment. Then $J_0^\star = J v_0^\star$.

Theorem 3 (Structural penalty margin)**.** *Under the delayed-cost model above, every covered policy π, including policies whose decisions may depend on previous segments, satisfies $J_P^\pi \leq J_0^\star - G_P n^\pi$. Consequently, $\rho_0 \leq [(L-1)\Delta_r - r_-]_+$. If $G_P > 0$, optimizing the relabeled objective over covered policies recovers a zero-cost policy with original return $J_0^\star$.*

Theorem 3 connects delayed-credit assignment to the event price. Longer delays allow more reward variation to accumulate before the cost event and therefore require a larger replacement penalty in the worst case. Conversely, when within-segment reward variation is limited, even a delayed cost signal can induce a clear preference for the safe continuation. Appendix B gives the proofs and a sufficient condition under which the practical data-derived scale $P_{\mathcal{D}}$ already yields $G_{P_{\mathcal{D}}} > 0$.

4.4 Why behavior regularization matters

The value separation in Lemma 2 is useful only if policy improvement can select the safer action. In offline RL, too little regularization can favor unsupported actions with unreliable values, while too much can prevent the policy from moving away from actions that lead to cost.

We make this tradeoff explicit in the delayed-cost model. Let $m_t(x)$ denote the conditional mean action in the behavior data (state x, phase t); write $D_t(x,a) = \|a - m_t(x)\|^2$. For squared behavior-cloning loss, replacing the sampled behavior action by $m_t(x)$ changes the conditional objective only by a term independent of the candidate action, so D_t captures the relevant geometry of regularizer.

Suppose that at every reachable viable state there is a covered viable action a_s with $D_t(x,a_s) \le d_s$, while every unsupported action satisfies $D_t(x,a) \ge d_o$, where $0 < d_s < d_o$. Thus a supported safe alternative is closer to the behavior than any action outside the covered region. Let $\widehat{Q}_t$ be the critic used for policy improvement. We assume that the safe witness is not underestimated by more than ϵ_s, that covered actions which make the delayed event unavoidable are not overestimated by more than ϵ_b, and that all action values obey the natural horizon-dependent upper bound:

$$\begin{aligned} \widehat{Q}_t(x,a_s) &\ge Tr_- - \epsilon_s, \\ \widehat{Q}_t(x,a_b) &\le (T-1)r_+ - P + \epsilon_b \quad \text{for every covered nonviable } a_b, \\ \widehat{Q}_t(x,a) &\le Tr_+ \qquad\qquad\qquad\quad \text{for every } a \in \mathcal{A}, \end{aligned} \tag{12}$$

where $T = L - t$. Lemma 2 gives the first two inequalities with zero error for the exact covered values.

Theorem 4 (Sufficient anchoring interval)**.** *Suppose the actor returns a τ-approximate maximizer of the regularized score $\widehat{Q}_t(x,a) - \beta D_t(x,a)$. Define*

$$\beta_{\mathrm{lo}} = \frac{L\Delta_r + \epsilon_s + \tau}{d_o - d_s}, \qquad \beta_{\mathrm{hi}} = \frac{G_P - \epsilon_s - \epsilon_b - \tau}{d_s}. \tag{13}$$

If $\beta_{\mathrm{lo}} < \beta_{\mathrm{hi}}$, then every $\beta \in (\beta_{\mathrm{lo}}, \beta_{\mathrm{hi}})$ makes the actor choose a covered viable action at every reachable viable state. The resulting policy therefore incurs no cost in the delayed-event construction.

The lower endpoint is the regularization needed to rule out unsupported actions despite critic error. The upper endpoint ensures that behavior regularization does not overwhelm the value advantage of a supported safer action. Increasing P increases G_P and enlarges this interval, while critic error narrows it. The theorem gives a sufficient condition for the stylized delayed-cost model, not a guarantee that ReBRAC remains inside this interval throughout training; proof is in Appendix B.6.

4.5 Why feasible complete trajectories are not necessary

Offline RL is often discussed in terms of coverage of the target policy by the behavior data (Chen and Jiang, 2019; Rashidinejad et al., 2021; Xie et al., 2021). This does not require the dataset to contain complete trajectories generated by the target policy. For RELABEL(COST), the distinction is especially important. A trajectory that violates the deployment budget may still contain many zero-cost transitions and supported safe continuations. The delayed-cost construction gives a simple example. Within each segment, the behavior policy can assign positive probability both to actions that preserve viability and to actions that eventually cause a cost event. As the number of segments in an episode grows, complete feasible episodes can become rare even though the local probability of observing a safe continuation remains unchanged.

Theorem 5 (Safe learning from entirely infeasible episodes)**.** *For every confidence level $\delta \in (0,1)$, there exists a finite deterministic offline RL instance, a behavior distribution, and a sufficiently large finite dataset such that, with probability at least $1-\delta$,*

1. *every recorded episode violates the deployment budget B, and*
2. *the transition data nevertheless identify a covered policy $\widehat{\pi}$ satisfying $n^{\widehat{\pi}} = J_C^{\widehat{\pi}} = 0$ and $J_R^{\widehat{\pi}} = J_0^\star$.*

The construction in Appendix C uses repeated delayed-cost segments. As the number of segments per episode increases, complete feasible episodes become exponentially rare, while the local probability of observing the safe continuation does not decrease. A finite-model identification argument can therefore recover a safe covered policy from data whose complete episodes are all infeasible.

Theorem 5 is an information-theoretic statement rather than a finite-sample guarantee for ReBRAC. It shows that complete feasible demonstrations are not inherently necessary for offline safe learning when the data contain enough local information to identify safe continuations. Section 5.3 tests the same question empirically by progressively removing low-cost trajectories.

Table 1: Normalized rewards ($R \uparrow$) and costs ($C \downarrow$); threshold 1 ($\kappa = 5$ Bullet, $\kappa = 10$ Safety Gym). Seed means (stds in Appendix D.1). **Bold**: safe ($C_{\text{norm}} \leq 1$). Gray: unsafe. **Blue**: best safe.

Task	BC-Safe		CPQ		COptiDICE		CDT		CAPS		FISOR		B2R		O3SRL		Relabel (Cost)	
	R↑	C↓	R↑	C↓	R↑	C↓	R↑	C↓	R↑	C↓	R↑	C↓	R↑	C↓	R↑	C↓	R↑	C↓
Bullet Gym ($\kappa = 5$)																		
BallRun	0.16	4.50	0.09	2.20	0.53	10.83	0.27	2.57	**0.07**	**0.00**	0.09	1.28	**0.30**	**0.93**	**0.25**	**0.00**	**0.27**	**0.00**
CarRun	**0.92**	**0.26**	**0.93**	**0.20**	**0.91**	**0.00**	**0.99**	**0.90**	**0.97**	**0.11**	**0.74**	**0.00**	**0.97**	**0.10**	**0.96**	**0.02**	**0.90**	**0.11**
DroneRun	0.41	1.62	0.29	2.35	0.68	15.02	**0.58**	**0.07**	0.41	5.70	0.31	2.52	**0.55**	**0.23**	**0.32**	**0.68**	**0.57**	**0.00**
AntRun	0.56	1.15	**0.03**	**0.05**	0.61	3.26	0.70	1.66	0.53	2.03	**0.43**	**0.27**	0.70	1.34	**0.33**	**0.14**	**0.69**	**0.43**
BallCircle	0.45	1.21	0.56	1.11	0.71	9.41	0.61	2.03	**0.33**	**0.01**	**0.32**	**0.00**	0.62	1.37	**0.62**	**0.06**	**0.72**	**0.01**
CarCircle	0.31	1.57	**0.71**	**0.00**	0.49	10.82	0.71	1.58	**0.40**	**0.03**	**0.37**	**0.00**	0.65	2.27	**0.64**	**0.11**	**0.66**	**0.00**
DroneCircle	0.50	1.22	**−0.21**	**0.70**	0.26	3.68	0.55	1.14	**0.36**	**0.00**	**0.48**	**0.17**	**0.47**	**0.72**	**0.49**	**0.23**	**0.59**	**0.01**
AntCircle	0.40	4.72	**0.00**	**0.00**	0.18	17.40	0.45	6.59	**0.33**	**0.00**	**0.24**	**0.04**	0.42	3.39	**0.48**	**0.00**	**0.86**	**0.08**
Safety Gym ($\kappa = 10$)																		
CarCircle1	**0.47**	**0.90**	0.48	4.32	**0.43**	**0.01**	0.63	1.14	0.46	2.61	**0.56**	**0.07**	0.06	1.92	0.50	4.15	**0.47**	**0.09**
CarCircle2	0.44	1.08	0.54	2.36	**0.53**	**0.42**	0.54	1.09	0.58	1.82	**0.59**	**0.80**	0.21	3.06	0.56	7.29	**0.47**	**0.30**
CarGoal1	**0.20**	**0.40**	0.67	4.41	0.40	2.50	0.62	3.12	0.27	1.15	0.42	1.70	0.47	1.81	0.59	4.14	**0.03**	**0.04**
CarGoal2	0.14	1.45	0.53	15.09	0.21	3.01	0.42	5.34	0.11	2.05	**0.05**	**0.50**	**0.12**	**1.00**	0.40	4.09	**−0.01**	**0.03**
PointCircle1	**0.30**	**0.26**	**0.46**	**0.73**	0.87	19.16	**0.54**	**0.54**	**0.31**	**0.94**	0.43	14.93	**0.49**	**0.48**	0.53	1.31	**0.59**	**0.01**
PointCircle2	0.42	2.10	0.44	1.20	0.85	29.36	0.59	2.05	**0.44**	**0.10**	0.76	18.02	0.57	1.01	**0.52**	**0.55**	**0.59**	**0.03**
PointGoal1	**0.22**	**0.91**	0.55	3.33	0.49	5.27	0.63	2.97	0.30	1.45	0.64	5.38	0.43	2.15	0.75	2.94	**−0.04**	**0.00**
PointGoal2	0.18	1.81	0.25	5.34	0.41	5.68	0.46	5.57	0.19	1.02	0.31	1.67	0.21	2.10	0.59	5.22	**0.00**	**0.00**
AntVelo	**0.92**	**0.33**	**−1.01**	**0.00**	1.00	11.82	**0.97**	**0.59**	**0.88**	**0.21**	**0.89**	**0.00**	**0.95**	**0.76**	**0.45**	**0.05**	**0.99**	**0.04**
HalfCheetahVelo	**0.86**	**0.30**	0.67	8.56	**0.63**	**0.00**	**0.98**	**0.65**	**0.86**	**0.27**	**0.89**	**0.00**	**0.93**	**0.10**	**0.93**	**0.00**	**0.94**	**0.00**
SwimmerVelo	**0.46**	**0.80**	0.20	2.82	0.59	16.80	0.66	1.26	0.45	2.17	**−0.02**	**0.23**	0.60	4.54	**0.05**	**0.00**	**0.51**	**0.00**

5 Experiments and Results

We ask whether RELABEL(COST) achieves competitive reward within cost budgets (Q1), how sensitive it is to penalty scale (Q2), and whether safe learning requires feasible trajectories (Q3).

Benchmarks and evaluation. We evaluate on DSRL (Liu et al., 2024), following FISOR (Zheng et al., 2024): $B = 5$ for BulletSafetyGym and $B = 10$ for Safety-Gymnasium. We report normalized return $R_{\text{norm}} = (R - R_{\min})/(R_{\max} - R_{\min})$ and cost $C_{\text{norm}} = C/B$. Safety requires $C_{\text{norm}} \leq 1$; safe policies are ranked by return, without preference for further cost reduction.

Offline RL backbone and baselines. RELABEL(COST) and all ablations use ReBRAC (Tarasov et al., 2023a) with identical task-specific configurations unless noted. Actor and critic regularization coefficients remain fixed within runs and are not rescaled with $P_{\mathcal{D}}$. Baselines are CPQ (Xu et al., 2022), COptiDICE (Lee et al., 2022), CDT (Liu et al., 2023), CAPS (Chemingui et al., 2025b), FISOR (Zheng et al., 2024), O3SRL (Chemingui et al., 2025a), and B2R (Su et al., 2026). Appendix E provides implementation details, hyperparameters, per-task results, and the actor coefficient's relation to β in Theorem 4.

Table 1 shows that RELABEL(COST) satisfies all 19 task budgets, achieves the highest feasible return on 9, and remains competitive elsewhere. On Safety-Gymnasium, cost stays below 5 despite a budget of 10. We next isolate relabeling's contribution within ReBRAC.

5.1 What does local relabeling contribute?

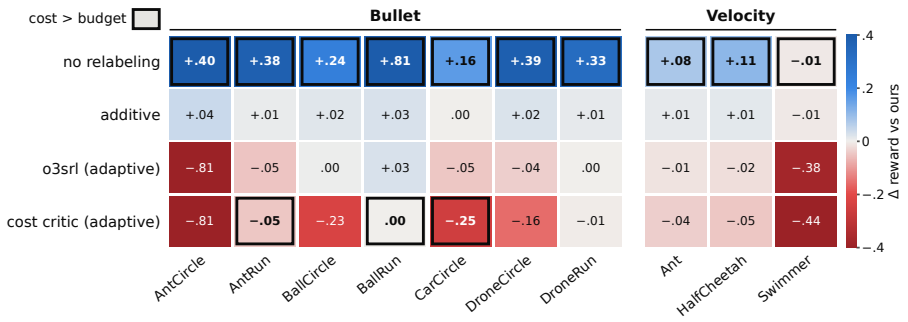


Figure 1: **Relabeling rules across tasks.** Cells show normalized-reward differences from RELABEL(COST) on matched tasks and seeds; outlined cells exceed the cost budget.

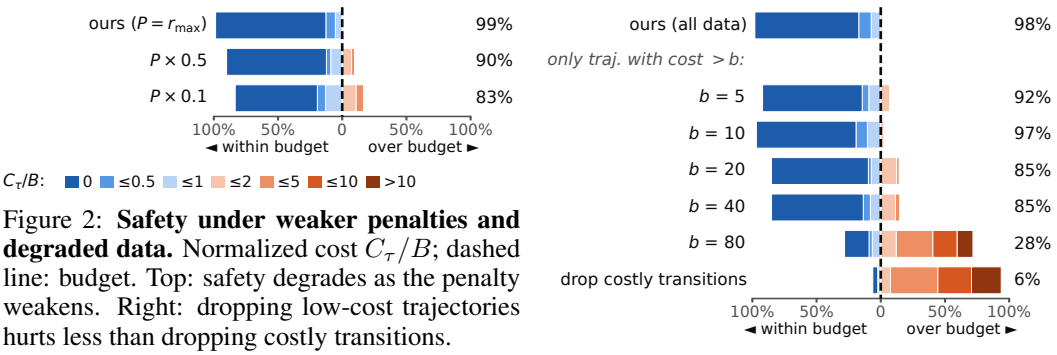


Figure 2: **Safety under weaker penalties and degraded data.** Normalized cost C_τ/B; dashed line: budget. Top: safety degrades as the penalty weakens. Right: dropping low-cost trajectories hurts less than dropping costly transitions.

Figure 1 varies only how safety information is incorporated into a fixed ReBRAC backbone. We make three observations. **1)** Without relabeling, all tasks violate the budget, showing that ReBRAC alone does not explain RELABEL(COST)'s safety. **2)** At the default scale, fixed additive relabeling is safe on all tasks with similar returns, suggesting that penalty strength matters more than replacement versus addition. **3)** an O3SRL-style adaptive penalty on ReBRAC remains safe but loses substantial reward on some tasks, while cost-critic relabeling yields lower average return and violates the budget on three tasks. Thus, neither adaptive penalties nor cost critics consistently improve over directly using observed transition-level costs.

5.2 How sensitive Relabel(cost) is to the penalty scale?

We test whether performance depends critically on the fixed data-derived penalty $P_\mathcal{D}$ by progressively weakening the event charge. Figure 2 shows that safety degrades as the penalty weakens: 99%, 90%, and 83% of episodes satisfy the budget for $P_\mathcal{D}$, $0.5P_\mathcal{D}$, and $0.1P_\mathcal{D}$, respectively, while normalized returns are 0.624, 0.655, and 0.624. Thus, $P_\mathcal{D}$ is not narrowly tuned: weaker penalties increase cost without reliably improving return, consistent with Section 4.1. For binary costs, replacement is also more robust to weaker penalties than addition, which can leave cost-bearing transitions positively rewarded when $r > \lambda P_\mathcal{D}$ (see Appendix for additional results).

5.3 What information does Relabel(cost) need from the data?

RELABEL(COST) acts on individual cost-bearing transitions rather than labeling complete trajectories as safe or unsafe. This raises a question of whether learning requires complete feasible demonstrations or the transition-level information contained in infeasible trajectories can suffice.

Removing feasible trajectories. For threshold b, we remove all trajectories with $C_\tau \leq b$, leaving no trajectory feasible under the filtering threshold. Surprisingly, safety remains largely intact through $b = 40$, with 85–97% of evaluation episodes satisfying the deployment budget and only moderate return loss (Figure 2). Performance deteriorates at $b = 80$, showing that arbitrary infeasible data are not sufficient. Consistent with Theorem 5, complete feasible trajectories are unnecessary while sufficient local transition information remains; retention and transition statistics are in Appendix.

Removing observed violations. We next ask whether the observed cost events themselves are necessary by removing all $c > 0$ transitions. This yields the highest normalized return (0.735), but only 6% of episodes satisfy the budget and 49% exceed five times the budget. Unlike removing complete low-cost trajectories, removing observed cost events eliminates direct negative supervision and largely destroys safety, although deletion also changes data coverage and the behavior distribution.

6 Conclusion

We studied RELABEL(COST), a simple offline safe RL method that replaces rewards on observed positive-cost transitions by a fixed negative value and then trains a behavior-regularized offline RL policy on the transformed dataset. Across the benchmark tasks, the method achieves low cost with strong reward performance, remains effective under moderate changes in the penalty scale, and can learn from datasets with no low-cost trajectories. The analysis explains how finite penalties control cost events, how delayed penalties propagate to earlier decisions, and how behavior regularization supports safe policy improvement. A natural next step is to incorporate the available budget more directly while retaining the simplicity of the method.

AI USE STATEMENT

We used generative AI tools to assist with polishing the manuscript, improving organization and presentation, and refining the presentation of experimental results. These tools were not used to formulate the research problem, develop the core technical contributions, or design the experiments. All AI-assisted content was reviewed and verified by the authors for accuracy, consistency, and alignment with the intended technical meaning. The authors take full responsibility for the final manuscript and all reported results.

ETHICS STATEMENT

This work studies methods for improving the safety of reinforcement learning agents through theoretical analysis and simulation-based experiments. The experiments involve no human participants or personal data. We are not aware of direct ethical concerns arising from the experimental procedures used in this work.

REFERENCES

Eitan Altman. *Constrained Markov Decision Processes*. Chapman and Hall/CRC, 1999. URL https://books.google.com/books?id=3X9S1NM2iOgC.

Yassine Chemingui, Aryan Deshwal, Alan Fern, Thanh Nguyen-Tang, and Jana Doppa. Online optimization for offline safe reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, pages 156036–156064, 2025a. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/cdc099a583d760335ee9b5ab21626a48-Abstract-Conference.html.

Yassine Chemingui, Aryan Deshwal, Honghao Wei, Alan Fern, and Jana Doppa. Constraint-adaptive policy switching for offline safe reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 15722–15730, 2025b. URL https://ojs.aaai.org/index.php/AAAI/article/view/33726.

Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1042–1051, 2019. URL https://proceedings.mlr.press/v97/chen19e.html.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://arxiv.org/abs/2106.06860.

Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(42):1437–1480, 2015. URL https://jmlr.org/papers/v16/garcia15a.html.

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

Junyu Guo, Zhi Zheng, Donghao Ying, Ming Jin, Shangding Gu, Costas J. Spanos, and Javad Lavaei. Don't trade off safety: Diffusion regularization for constrained offline rl. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/ea82157b51a522a4c2eb760755af76c9-Abstract-Conference.html.

Matthew Thomas Jackson, Uljad Berdica, Jarek Liesen, Shimon Whiteson, and Jakob Nicolaus Foerster. A clean slate for offline reinforcement learning, 2025. URL https://arxiv.org/abs/2504.11453.

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

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=68n2s9ZJWF8.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 1179–1191, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/0d2b2061826a5df3221116a5085a6052-Abstract.html.

Jongmin Lee, Cosmin Paduraru, Daniel J. Mankowitz, Nicolas Heess, Doina Precup, Kee-Eung Kim, and Arthur Guez. Coptidice: Offline constrained reinforcement learning via stationary distribution correction estimation. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2204.08957.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020. URL https://arxiv.org/abs/2005.01643.

Zifan Liu, Xinran Li, and Jun Zhang. C2iql: Constraint-conditioned implicit q-learning for safe offline reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 38827–38841, 2025. URL https://proceedings.mlr.press/v267/liu25ai.html.

Zuxin Liu, Zijian Guo, Yihang Yao, Zhepeng Cen, Wenhao Yu, Tingnan Zhang, and Ding Zhao. Constrained decision transformer for offline safe reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 21611–21630, 2023. URL https://proceedings.mlr.press/v202/liu23m.html.

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*, 1(12):1–29, 2024. URL https://openreview.net/forum?id=ajf5PE3Z7w.

Paria Rashidinejad, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart J. Russell. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://papers.nips.cc/paper_files/paper/2021/hash/60ce36723c17bbac504f2ef4c8a46995-Abstract.html.

Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*, 2019.

Huikang Su, Dengyun Peng, Zifeng Zhuang, Yuhan Liu, Qiguang Chen, Donglin Wang, and Qinghe Liu. Boundary-to-region supervision for offline safe reinforcement learning. *Advances in Neural Information Processing Systems*, 38:57917–57946, 2026.

Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023a. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/26cce1e512793f2072fd27c391e04652-Abstract.html.

Denis Tarasov, Alexander Nikulin, Dmitry Akimov, Vladislav Kurenkov, and Sergey Kolesnikov. CORL: Research-oriented deep offline reinforcement learning library. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, 2023b.

Tengyang Xie, Ching-An Cheng, Nan Jiang, Paul Mineiro, and Alekh Agarwal. Bellman-consistent pessimism for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://arxiv.org/abs/2106.06926.

Haoran Xu, Xianyuan Zhan, and Xiangyu Zhu. Constraints penalized q-learning for safe offline reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8753–8760, 2022. URL https://ojs.aaai.org/index.php/AAAI/article/view/20855.

Yihang Yao, Zhepeng Cen, Wenhao Ding, Haohong Lin, Shiqi Liu, Tingnan Zhang, Wenhao Yu, and Ding Zhao. Oasis: Conditional distribution shaping for offline safe reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/8f75af4704feac629a560f4ad6b67cef-Abstract-Conference.html.

Yinan Zheng, Jianxiong Li, Dongjie Yu, Yujie Yang, Shengbo Eben Li, Xianyuan Zhan, and Jingjing Liu. Safe offline reinforcement learning with feasibility-guided diffusion model. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=j5JvZCaDM0.

A Additional Theory for Local Cost Relabeling

A.1 Proof of Theorem 1

For any policy π,

$$J_P^\pi = J_R^\pi - \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{H-1}(P + r_t)v_t\right]. \tag{A.1}$$

Since $r_t \geq r_-$ whenever $v_t = 1$ and $J_R^\pi \leq Hr_+$,

$$J_P^\pi \leq Hr_+ - (P + r_-)n^\pi. \tag{A.2}$$

Because $\pi_0 \in \Pi_0$, it has $n^{\pi_0} = 0$ and therefore

$$J_P^{\pi_0} = J_R^{\pi_0}. \tag{A.3}$$

The assumed comparison $J_P^{\widehat{\pi}} \geq J_P^{\pi_0} - \epsilon_{\mathrm{RL}}$ and Equation A.2 then give

$$Hr_+ - (P + r_-)n^{\widehat{\pi}} \geq J_R^{\pi_0} - \epsilon_{\mathrm{RL}}. \tag{A.4}$$

Rearranging,

$$n^{\widehat{\pi}} \leq \frac{Hr_+ - J_R^{\pi_0} + \epsilon_{\mathrm{RL}}}{P + r_-}. \tag{A.5}$$

Finally, Lemma 1 gives $J_C^{\widehat{\pi}} \leq c_{\max} n^{\widehat{\pi}}$ and $p_B^{\widehat{\pi}} \leq \min\{1, n^{\widehat{\pi}}/k_B\}$. Substituting Equation A.5 proves Equation 6.

A.2 Proof of Theorem 2

For any policy with $n^\pi > 0$, the definition in Equation 7 implies

$$A^\pi \leq J_0^\star + \rho_0 n^\pi. \tag{A.6}$$

The same inequality also holds for $n^\pi = 0$, since then $A^\pi = J_R^\pi \leq J_0^\star$. Therefore, for every $\pi \in \Pi$,

$$J_P^\pi = A^\pi - Pn^\pi \leq J_0^\star - (P - \rho_0)n^\pi, \tag{A.7}$$

which proves Equation 8.

If $P > \rho_0$, every policy with $n^\pi > 0$ has $J_P^\pi < J_0^\star$. On the other hand, zero-cost policies satisfy $J_P^\pi = J_R^\pi$, so their supremum under the relabeled objective is $J_0^\star$. Hence $\sup_{\pi\in\Pi} J_P^\pi = J_0^\star$, and any maximizing policy must have $n^\pi = 0$ and original return $J_0^\star$.

If $0 \leq P < \rho_0$, the definition of the supremum implies that there exists a policy with $n^\pi > 0$ such that

$$\frac{A^\pi - J_0^\star}{n^\pi} > P. \tag{A.8}$$

For this policy, $J_P^\pi = A^\pi - Pn^\pi > J_0^\star$. When $P = \rho_0$, the strict separation disappears, so positive-cost policies may tie with the zero-cost optimum. □

A.3 A local sufficient bound on the event price

The event price in Equation 7 is global and is generally unknown. A more structured sufficient bound can be obtained by comparing individual deviations from a zero-cost reference policy.

Fix a zero-cost reference policy π_0. Let $Q_A^{\pi_0}(s,a)$ and $V_A^{\pi_0}(s)$ denote its finite-horizon values under the reward $r_t(1 - v_t)$, with time included in the state. Let $N^{\pi_0}(s,a)$ denote the expected number of future cost events after taking action a in state s and then following π_0, and let $N^{\pi_0}(s)$ be the corresponding value when following π_0 immediately. Define

$$\Delta_A(s,a) = Q_A^{\pi_0}(s,a) - V_A^{\pi_0}(s), \qquad \Delta_N(s,a) = N^{\pi_0}(s,a) - N^{\pi_0}(s). \tag{A.9}$$

Assume that on all state–action pairs visited by policies in Π, $\Delta_N(s,a) \geq 0$, and that $\Delta_A(s,a) \leq 0$ whenever $\Delta_N(s,a) = 0$. Thus, relative to π_0, a deviation cannot reduce the expected number of

future cost events, and any deviation that leaves the event count unchanged cannot improve the zero-cost reward.

Define

$$\rho_{\mathrm{loc}} = \sup_{\Delta_N(s,a)>0} \frac{[\Delta_A(s,a)]_+}{\Delta_N(s,a)}, \tag{A.10}$$

again taking the supremum of an empty set to be zero.

Proposition 2 (Local sufficient event price)**.** *Under the conditions above,*

$$\rho_0 \le \rho_{\mathrm{loc}}. \tag{A.11}$$

Consequently, every $P > \rho_{\mathrm{loc}}$ also satisfies the exact-penalty conclusion of Theorem 2.

Proof. The assumptions imply $\Delta_A(s,a) \le \rho_{\mathrm{loc}}\Delta_N(s,a)$ on all relevant state–action pairs. Applying the finite-horizon performance-difference identity to the reward $r(1-v)$ gives

$$A^\pi - A^{\pi_0} = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{H-1} \Delta_A(s_t,a_t)\right]. \tag{A.12}$$

Applying the same identity to the event indicator v_t, and using $n^{\pi_0} = 0$, gives

$$n^\pi = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{H-1} \Delta_N(s_t,a_t)\right]. \tag{A.13}$$

Therefore $A^\pi - A^{\pi_0} \le \rho_{\mathrm{loc}} n^\pi$. Since $A^{\pi_0} = J_R^{\pi_0} \le J_0^\star$, we obtain

$$A^\pi - J_0^\star \le \rho_{\mathrm{loc}} n^\pi. \tag{A.14}$$

Taking the supremum over policies with $n^\pi > 0$ proves $\rho_0 \le \rho_{\mathrm{loc}}$. □

The local bound is not needed by the algorithm. Its purpose is to show how the global event price can arise from a sequence of local reward-versus-safety tradeoffs. In particular, the denominator in Equation A.10 measures the additional expected number of future cost events caused by a deviation, while the numerator measures the corresponding gain in zero-cost reward.

A.4 Proof of Proposition 1

Let π be any policy satisfying $J_C^\pi \le B$. Since every positive-cost transition has cost at least $c_{\min}$,

$$c_{\min} n^\pi \le J_C^\pi \le B. \tag{A.15}$$

Therefore,

$$n^\pi \le \frac{B}{c_{\min}}. \tag{A.16}$$

Recall that

$$A^\pi = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{H-1} r_t(1-v_t)\right]. \tag{A.17}$$

Hence the original return decomposes as

$$J_R^\pi = A^\pi + \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{H-1} r_t v_t\right]. \tag{A.18}$$

By the definition of the event price in Equation 7,

$$A^\pi \le J_0^\star + \rho_0 n^\pi. \tag{A.19}$$

The reward collected on positive-cost transitions satisfies

$$\mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{H-1} r_t v_t\right] \le r_+^+ n^\pi, \tag{A.20}$$

where $r_+^+ = \max\{r_+, 0\}$.

Combining Equations A.18, A.19, and A.20 gives

$$J_R^\pi - J_0^\star \le \big(\rho_0 + r_+^+\big) n^\pi. \tag{A.21}$$

Using Equation A.16,

$$J_R^\pi - J_0^\star \le \frac{B}{c_{\min}} \big(\rho_0 + r_+^+\big). \tag{A.22}$$

Since this holds for every policy satisfying $J_C^\pi \le B$, taking the supremum over all budget-feasible policies yields

$$J_B^\star - J_0^\star \le \frac{B}{c_{\min}} \big(\rho_0 + r_+^+\big). \tag{A.23}$$

The lower bound $J_B^\star \ge J_0^\star$ follows because every zero-cost policy is feasible for every $B \ge 0$. For binary unit costs, $c_{\min} = 1$, giving

$$J_B^\star - J_0^\star \le B\big(\rho_0 + r_+^+\big). \tag{A.24}$$

B Additional Results for Delayed Cost Events

B.1 Covered dynamic programming

For the deterministic delayed-cost model in Section 4.3, let $Q_t^{\mathcal{C}}$ and $V_t^{\mathcal{C}}$ denote the optimal values when actions are restricted to the covered set $\mathcal{C}$. They satisfy

$$Q_t^{\mathcal{C}}(x, a) = \widetilde{r}_{P,t}(x, a) + V_{t+1}^{\mathcal{C}}(f_t(x, a)), \qquad a \in \mathcal{C}, \tag{B.1}$$

and

$$V_t^{\mathcal{C}}(x) = \max_{a \in \mathcal{C}} Q_t^{\mathcal{C}}(x, a), \qquad V_L^{\mathcal{C}} = 0. \tag{B.2}$$

The viability sets in Equation 10 characterize whether the terminal cost event can still be avoided while remaining inside the covered action set. In particular, if $x \in \mathcal{K}_t$ and $f_t(x, a) \notin \mathcal{K}_{t+1}$, then the terminal event is unavoidable after taking a and restricting subsequent actions to $\mathcal{C}$.

B.2 Proof of Lemma 2

Because $x \in \mathcal{K}_t$ and $f_t(x, a_s) \in \mathcal{K}_{t+1}$, there exists a covered continuation after a_s that reaches G and therefore incurs no positive-cost event. There are $T = L - t$ rewards remaining in the segment, each at least r_-. Hence

$$Q_t^{\mathcal{C}}(x, a_s) \ge T r_-. \tag{B.3}$$

By contrast, $f_t(x, a_b) \notin \mathcal{K}_{t+1}$ means that no covered continuation after a_b can reach G. The terminal positive-cost event is therefore unavoidable. At most $T - 1$ ordinary rewards can be collected before the final transition, each bounded above by r_+, while the terminal relabeled reward is $-P$. Thus

$$Q_t^{\mathcal{C}}(x, a_b) \le (T - 1) r_+ - P. \tag{B.4}$$

Subtracting Equation B.4 from Equation B.3 gives

$$Q_t^{\mathcal{C}}(x, a_s) - Q_t^{\mathcal{C}}(x, a_b) \ge P + r_- - (T - 1)\Delta_r \ge P + r_- - (L - 1)\Delta_r = G_P, \tag{B.5}$$

which proves Lemma 2.

B.3 Proof of Theorem 3

Consider one segment. If the segment incurs no positive-cost event, its relabeled return equals its original return and is at most $v_0^\star$, the optimal covered zero-cost return for one segment.

If the segment incurs a positive-cost event, then its first $L - 1$ rewards are each at most r_+ and its final relabeled reward is $-P$. Its relabeled return is therefore at most $(L - 1)r_+ - P$. Since a zero-cost covered continuation exists and receives at least r_- at each step, $v_0^\star \ge L r_-$. Hence

$$(L - 1)r_+ - P \le v_0^\star - G_P. \tag{B.6}$$

Let Z_j indicate whether segment j incurs a positive-cost event. Conditional on any history before segment j, its relabeled return is therefore at most $v_0^\star - G_P Z_j$. This remains true even when the policy for the current segment depends on the outcomes of previous segments. Summing over the J segments and taking expectations yields

$$J_P^\pi \le J v_0^\star - G_P \mathbb{E}_\pi \left[\sum_{j=1}^{J} Z_j \right]. \tag{B.7}$$

Each segment contains at most one positive-cost event, so $\mathbb{E}_\pi[\sum_j Z_j] = n^\pi$. Because the environment resets after every segment, the optimal zero-cost return is $J_0^\star = J v_0^\star$. This yields

$$J_P^\pi \le J_0^\star - G_P n^\pi. \tag{B.8}$$

Using $G_P = P + r_- - (L-1)\Delta_r$, Equation B.8 implies

$$A^\pi - J_0^\star \le \big((L-1)\Delta_r - r_-\big) n^\pi. \tag{B.9}$$

Taking the positive part, dividing by n^π, and using the definition of ρ_0 in Equation 7 gives

$$\rho_0 \le [(L-1)\Delta_r - r_-]_+ . \tag{B.10}$$

Finally, if $G_P > 0$, Equation B.8 is a positive penalty margin, so Theorem 2 implies exact recovery of a zero-cost optimum among covered policies.

B.4 When the data-derived scale is sufficient

The practical algorithm uses $P_\mathcal{D} = \max_{i \in \mathcal{D}} |r_i|$ rather than the unknown event price. The following special case gives a structural condition under which this data-derived scale is already large enough.

Corollary 1 (Sufficiency of the data-derived scale)**.** *Suppose segments have fixed length L, there is no early termination within a segment, and $r_- > 0$. If*

$$2r_- - (L-1)\Delta_r > 0, \tag{B.11}$$

then every nonempty dataset satisfies $P_\mathcal{D} \ge r_-$ and therefore $G_{P_\mathcal{D}} > 0$. Consequently, optimizing the relabeled objective over covered policies recovers a zero-cost policy with original return $J_0^\star$.

Proof. Because every observed reward is at least $r_- > 0$ and the dataset is nonempty, $P_\mathcal{D} = \max_{i \in \mathcal{D}} |r_i| \ge r_-$. Therefore

$$G_{P_\mathcal{D}} = P_\mathcal{D} + r_- - (L-1)\Delta_r \ge 2r_- - (L-1)\Delta_r. \tag{B.12}$$

Condition B.11 makes this quantity positive, and the result follows from Theorem 3. □

This corollary is a sufficient structural condition, not a general certificate for the practical choice $P_\mathcal{D}$. Outside this regime, the event-price characterization in Theorem 2 remains the relevant population-level condition.

B.5 Behavior regularization and the conditional mean

Let $a_\mathcal{D}$ denote the recorded behavior action at state x and phase t, and let

$$m_t(x) = \mathbb{E}[a_\mathcal{D} \mid x, t]. \tag{B.13}$$

For any candidate action a,

$$\mathbb{E}\big[\|a - a_\mathcal{D}\|^2 \mid x, t\big] = \|a - m_t(x)\|^2 + \mathbb{E}\big[\|a_\mathcal{D} - m_t(x)\|^2 \mid x, t\big]. \tag{B.14}$$

The second term does not depend on a. Thus, for the purpose of policy improvement at (x, t), squared behavior-cloning regularization is equivalent to the penalty

$$D_t(x, a) = \|a - m_t(x)\|^2. \tag{B.15}$$

This identity motivates the geometric assumptions in Theorem 4. A covered viable witness a_s lies within squared distance d_s of the behavior mean, while every unsupported action lies at squared distance at least $d_o > d_s$.

B.6 Proof of Theorem 4

Fix a reachable viable state x at phase t and let $T = L - t$. The covered viable witness a_s has regularized score at least

$$Tr_- - \epsilon_s - \beta d_s. \tag{B.16}$$

Consider first any covered nonviable action a_b. By Equation 12, its value is at most $(T-1)r_+ - P + \epsilon_b$, and its regularization penalty is nonnegative. Therefore the safe witness exceeds its score by at least

$$\begin{aligned} &Tr_- - \epsilon_s - \beta d_s - \big((T-1)r_+ - P + \epsilon_b\big) \\ &\quad = P + r_- - (T-1)\Delta_r - \epsilon_s - \epsilon_b - \beta d_s \\ &\quad \geq G_P - \epsilon_s - \epsilon_b - \beta d_s. \end{aligned} \tag{B.17}$$

The condition $\beta < \beta_{\mathrm{hi}}$ in Equation 13 makes the final expression larger than τ. Hence a τ-approximate maximizer cannot choose a covered nonviable action.

Now consider an unsupported action a. Its squared distance from the behavioral mean is at least d_o, while Equation 12 gives $\widehat{Q}_t(x,a) \leq Tr_+$. The safe witness therefore exceeds its regularized score by at least

$$\begin{aligned} &Tr_- - \epsilon_s - \beta d_s - \big(Tr_+ - \beta d_o\big) \\ &\quad = \beta(d_o - d_s) - T\Delta_r - \epsilon_s \\ &\quad \geq \beta(d_o - d_s) - L\Delta_r - \epsilon_s. \end{aligned} \tag{B.18}$$

The condition $\beta > \beta_{\mathrm{lo}}$ makes this gap larger than τ. Thus a τ-approximate maximizer cannot choose an unsupported action either.

Every action returned by the actor is therefore both covered and viable. Since the initial state lies in $\mathcal{K}_0$, induction using the definition of the viability sets in Equation 10 keeps the trajectory in $\mathcal{K}_t$ until the terminal state lies in G. Equation 9 then implies that no positive-cost event occurs. Repeating the same argument after each segment reset proves the claim.

C Constructive Separation with Infeasible Episodes

This section proves Theorem 5. The construction separates two notions of coverage. Local transition coverage remains bounded away from zero, so the dynamics and safe continuation can be identified from sufficiently many observed segments. At the same time, the probability that a complete behavior episode satisfies the deployment budget decreases exponentially in the number of segments grouped into that episode.

We first give a deterministic continuous-control family and then place a finite model class over that family. The resulting identification argument yields a finite dataset for which every recorded episode is infeasible while the optimal covered zero-cost policy is still identified.

C.1 A continuous delayed-cost family

Let $\mathcal{A}$ and $\mathcal{C}$ be compact intervals with $\mathcal{C} \subset \operatorname{int}\mathcal{A}$. Each segment starts from $x_0 = x_{\max} > 0$ and evolves according to

$$x_{t+1} = x_t \psi(a_t), \qquad G = [\rho, x_{\max}], \qquad 0 < \rho < x_{\max}. \tag{C.1}$$

The continuous map $\psi : \mathcal{A} \to (0,1]$ equals one on a nondegenerate interval $I_s \subset \operatorname{int}\mathcal{C}$ and is at most $q < \rho/x_{\max}$ on a nondegenerate interval $I_b \subset \operatorname{int}\mathcal{C}$. The behavior policy draws actions independently from a full-support distribution on $\mathcal{C}$ at every phase. Write $p_s = \mathbb{P}(a \in I_s) > 0$ and $p_b = \mathbb{P}(a \in I_b) > 0$.

Repeated actions in I_s preserve the state and therefore form a zero-cost continuation. An action in I_b moves the state below ρ in one step, after which the terminal event is unavoidable. More generally,

$$x_t = x_{\max} \prod_{j<t} \psi(a_j), \tag{C.2}$$

and the viability sets in Equation 10 reduce to $\mathcal{K}_t = [\rho, x_{\max}]$.

Let q_b denote the probability that one behavior segment incurs a cost event. Then

$$1 - (1 - p_b)^L \le q_b \le 1 - p_s^L < 1. \tag{C.3}$$

The lower bound follows because any action in I_b causes failure, while the upper bound follows because L actions in I_s guarantee success.

The key feature is that increasing the number of segments per episode does not reduce local coverage. At phase t, the behavior policy reaches the safe plateau state $x_t = x_{\max}$ with probability at least p_s^t, independently of the number of segments used to form an episode. Complete-episode feasibility, by contrast, deteriorates as more segments are concatenated.

C.2 Finite-model identification

Let $\mathfrak{M}$ be a finite class of deterministic templates (ψ_j, ρ_k) containing the true environment. The reward function is common to all templates and known to the learner, the covered action set $\mathcal{C}$ is known, and observations are noiseless.

Two templates are *covered-equivalent* if they predict the same successor and relabeled reward at every true reachable covered state-action pair. For an inequivalent template M, let

$$\eta_{\mathsf{M}} = \mathbb{P}_{\mathrm{beh}}\left\{\text{one fresh segment contains a record refuting } \mathsf{M}\right\}, \qquad \eta = \min_{\mathsf{M} \text{ inequivalent}} \eta_{\mathsf{M}}. \tag{C.4}$$

Under the continuity and full-support conditions above, $\eta > 0$ whenever an inequivalent template exists.

The learner retains and relabels every transition, chooses any template consistent with the observed successors and relabeled rewards, computes its optimal covered values using Equation 11, and returns a covered maximizing policy.

Theorem 6 (Quantitative infeasible-data separation)**.** *Collect M independent behavior episodes, each consisting of J independent segments, for a total of $N = MJ$ segments. Let $c_{\mathrm{e}} > 0$ be the cost of a failed segment and set $b = B/(Jc_{\mathrm{e}})$. Assume $b < q_b$ and choose P so that $G_P > 0$. Then, with probability at least*

$$1 - (|\mathfrak{M}| - 1)e^{-MJ\eta} - M \exp\left\{-J \ \mathrm{kl}(b \| q_b)\right\}, \tag{C.5}$$

every recorded episode has cumulative cost greater than B, while the learner returns a policy $\widehat{\pi}$ satisfying

$$n^{\widehat{\pi}} = J_C^{\widehat{\pi}} = 0, \qquad J_R^{\widehat{\pi}} = J_0^\star. \tag{C.6}$$

Here kl *denotes binary relative entropy.*

Proof. An inequivalent template survives one independent segment with probability at most $1 - \eta$, and therefore survives all MJ observed segments with probability at most $e^{-MJ\eta}$. A union bound over the at most $|\mathfrak{M}| - 1$ inequivalent templates shows that, except on an event of probability $(|\mathfrak{M}|-1)e^{-MJ\eta}$, every template consistent with the data is covered-equivalent to the truth. Covered planning therefore uses the correct covered values, and Theorem 3 gives zero cost and return $J_0^\star$.

The number of failed segments in a behavior episode is distributed as $\mathrm{Binomial}(J, q_b)$. Since its cumulative cost is c_{e} times this count,

$$\mathbb{P}(C_\tau \le B) \le \exp\left\{-J \ \mathrm{kl}(b \| q_b)\right\} \tag{C.7}$$

for $b < q_b$ by the lower-tail Chernoff bound. A union bound over the M episodes and the identification error yields Equation C.5. Independence between the identification event and the episode-feasibility events is not required. □

For fixed segment length and model class, identification improves with the total number of observed segments MJ, whereas complete-episode feasibility worsens as the number J of segments grouped into each episode grows. This is the source of the separation. Local transition evidence can become increasingly informative even while complete feasible demonstrations become increasingly unlikely.

Table 2: Main results for normalized rewards and costs. The normalized cost threshold is 1 ($\kappa = 5$ or 10). The ↑ symbol denotes that higher rewards are better. The ↓ symbol denotes that lower costs (up to threshold 1) are better. Each cell shows reward or cost with standard deviation. **Bold**: safe agents ($C_{\text{norm}} \leq 1$). Gray: unsafe agents. **Blue bold**: best safe result ($C_{\text{norm}} \leq 1$).

Tasks	Metric	BC Safe	CPQ	COptiDICE	CDT	CAPS	FISOR	B2R	O3SRL	Relabel(Cost)
Bullet Gym Tasks: $\kappa = 5$										
BallRun	Reward ↑	0.16 ± 0.11	0.09 ± 0.26	0.53 ± 0.10	0.27 ± 0.09	**0.07 ± 0.05**	0.09 ± 0.07	**0.30 ± 0.02**	**0.25 ± 0.03**	**0.27 ± 0.02**
	Cost ↓	4.50 ± 3.38	2.20 ± 2.88	10.83 ± 1.68	2.57 ± 3.23	**0.00 ± 0.00**	1.28 ± 1.70	**0.93 ± 1.61**	**0.00 ± 0.00**	**0.00 ± 0.00**
CarRun	Reward ↑	**0.92 ± 0.01**	**0.93 ± 0.01**	**0.91 ± 0.04**	**0.99 ± 0.00**	**0.97 ± 0.00**	**0.74 ± 0.01**	**0.97 ± 0.01**	**0.96 ± 0.01**	**0.90 ± 0.01**
	Cost ↓	**0.26 ± 0.23**	**0.20 ± 0.18**	**0.00 ± 0.00**	**0.90 ± 0.34**	**0.11 ± 0.17**	**0.00 ± 0.00**	**0.10 ± 0.17**	**0.02 ± 0.03**	**0.11 ± 0.15**
DroneRun	Reward ↑	0.41 ± 0.23	0.29 ± 0.11	0.68 ± 0.01	**0.58 ± 0.00**	0.41 ± 0.06	0.31 ± 0.04	**0.55 ± 0.01**	**0.32 ± 0.05**	**0.57 ± 0.02**
	Cost ↓	1.62 ± 1.73	2.35 ± 4.08	15.02 ± 0.10	**0.07 ± 0.07**	5.70 ± 3.08	2.52 ± 1.10	**0.23 ± 0.39**	**0.68 ± 1.18**	**0.00 ± 0.00**
AntRun	Reward ↑	0.56 ± 0.02	**0.03 ± 0.05**	0.61 ± 0.01	0.70 ± 0.03	0.53 ± 0.13	**0.43 ± 0.02**	0.70 ± 0.03	**0.33 ± 0.13**	**0.69 ± 0.01**
	Cost ↓	1.15 ± 0.47	**0.05 ± 0.08**	3.26 ± 1.39	1.66 ± 0.24	2.03 ± 2.17	**0.27 ± 0.15**	1.34 ± 0.16	**0.14 ± 0.10**	**0.43 ± 0.26**
BallCircle	Reward ↑	0.45 ± 0.03	0.56 ± 0.10	0.71 ± 0.01	0.61 ± 0.17	**0.33 ± 0.02**	**0.32 ± 0.05**	0.62 ± 0.04	**0.62 ± 0.01**	**0.72 ± 0.01**
	Cost ↓	1.21 ± 0.23	1.11 ± 1.92	9.41 ± 0.69	2.03 ± 1.39	**0.01 ± 0.02**	**0.00 ± 0.00**	1.37 ± 1.00	**0.06 ± 0.07**	**0.01 ± 0.01**
CarCircle	Reward ↑	0.31 ± 0.06	**0.71 ± 0.02**	0.49 ± 0.01	0.71 ± 0.01	**0.40 ± 0.03**	**0.37 ± 0.02**	0.65 ± 0.05	**0.64 ± 0.03**	**0.66 ± 0.03**
	Cost ↓	1.57 ± 0.07	**0.00 ± 0.00**	10.82 ± 1.28	1.58 ± 0.57	**0.03 ± 0.03**	**0.00 ± 0.00**	2.27 ± 1.16	**0.11 ± 0.16**	**0.00 ± 0.00**
DroneCircle	Reward ↑	0.50 ± 0.01	**-0.21 ± 0.04**	0.26 ± 0.02	0.55 ± 0.01	**0.36 ± 0.02**	**0.48 ± 0.01**	**0.47 ± 0.02**	**0.49 ± 0.07**	**0.59 ± 0.02**
	Cost ↓	1.22 ± 0.31	**0.70 ± 0.41**	3.68 ± 0.51	1.14 ± 0.06	**0.00 ± 0.00**	**0.17 ± 0.15**	**0.72 ± 0.85**	**0.23 ± 0.34**	**0.01 ± 0.02**
AntCircle	Reward ↑	0.40 ± 0.02	**0.00 ± 0.00**	0.18 ± 0.02	0.45 ± 0.05	**0.33 ± 0.05**	**0.24 ± 0.02**	0.42 ± 0.04	**0.48 ± 0.06**	**0.86 ± 0.03**
	Cost ↓	4.72 ± 0.87	**0.00 ± 0.00**	17.40 ± 0.36	6.59 ± 1.42	**0.00 ± 0.00**	**0.04 ± 0.08**	3.39 ± 0.72	**0.00 ± 0.00**	**0.08 ± 0.12**
Safety Gym Tasks: $\kappa = 10$										
CarCircle1	Reward ↑	**0.47 ± 0.03**	0.48 ± 0.12	**0.43 ± 0.02**	0.63 ± 0.03	0.46 ± 0.10	**0.56 ± 0.02**	0.06 ± 0.13	0.50 ± 0.05	**0.47 ± 0.03**
	Cost ↓	**0.90 ± 0.29**	4.32 ± 1.12	**0.01 ± 0.02**	1.14 ± 0.06	2.61 ± 0.39	**0.07 ± 0.04**	1.92 ± 1.39	4.15 ± 1.23	**0.09 ± 0.15**
CarCircle2	Reward ↑	0.44 ± 0.10	0.54 ± 0.05	**0.53 ± 0.02**	0.54 ± 0.09	0.58 ± 0.02	**0.59 ± 0.02**	0.21 ± 0.01	0.56 ± 0.00	**0.47 ± 0.05**
	Cost ↓	1.08 ± 1.03	2.36 ± 2.14	**0.42 ± 0.17**	1.09 ± 0.73	1.82 ± 0.58	**0.80 ± 0.29**	3.06 ± 2.08	7.29 ± 2.07	**0.30 ± 0.24**
CarGoal1	Reward ↑	**0.20 ± 0.09**	0.67 ± 0.07	0.40 ± 0.03	0.62 ± 0.04	0.27 ± 0.00	0.42 ± 0.07	0.47 ± 0.06	0.59 ± 0.10	**0.03 ± 0.03**
	Cost ↓	**0.40 ± 0.12**	4.41 ± 0.46	2.50 ± 0.36	3.12 ± 0.69	1.15 ± 0.10	1.70 ± 0.58	1.81 ± 0.60	4.14 ± 0.91	**0.04 ± 0.06**
CarGoal2	Reward ↑	0.14 ± 0.02	0.53 ± 0.14	0.21 ± 0.10	0.42 ± 0.06	0.11 ± 0.06	**0.05 ± 0.01**	**0.12 ± 0.03**	0.40 ± 0.30	**-0.01 ± 0.03**
	Cost ↓	1.45 ± 0.60	15.09 ± 1.91	3.01 ± 1.45	5.34 ± 1.25	2.05 ± 2.23	**0.50 ± 0.71**	**1.00 ± 0.46**	4.09 ± 3.40	**0.03 ± 0.02**
PointCircle1	Reward ↑	**0.30 ± 0.11**	**0.46 ± 0.04**	0.87 ± 0.01	**0.54 ± 0.01**	**0.31 ± 0.05**	0.43 ± 0.05	**0.49 ± 0.01**	0.53 ± 0.02	**0.59 ± 0.01**
	Cost ↓	**0.26 ± 0.24**	**0.73 ± 1.20**	19.16 ± 0.38	**0.54 ± 0.61**	**0.94 ± 1.53**	14.93 ± 4.24	**0.48 ± 0.69**	1.31 ± 0.62	**0.01 ± 0.02**
PointCircle2	Reward ↑	0.42 ± 0.07	0.44 ± 0.05	0.85 ± 0.01	0.59 ± 0.02	**0.44 ± 0.06**	0.76 ± 0.05	0.57 ± 0.03	**0.52 ± 0.04**	**0.59 ± 0.01**
	Cost ↓	2.10 ± 0.37	1.20 ± 1.28	29.36 ± 1.23	2.05 ± 0.93	**0.10 ± 0.09**	18.02 ± 4.17	1.01 ± 0.52	**0.55 ± 0.41**	**0.03 ± 0.06**
PointGoal1	Reward ↑	**0.22 ± 0.02**	0.55 ± 0.11	0.49 ± 0.02	0.63 ± 0.08	0.30 ± 0.09	0.64 ± 0.03	0.43 ± 0.09	0.75 ± 0.01	**-0.04 ± 0.02**
	Cost ↓	**0.91 ± 0.42**	3.33 ± 1.68	5.27 ± 0.91	2.97 ± 0.82	1.45 ± 0.74	5.38 ± 1.05	2.15 ± 0.55	2.94 ± 1.06	**0.00 ± 0.00**
PointGoal2	Reward ↑	0.18 ± 0.01	0.25 ± 0.13	0.41 ± 0.06	0.46 ± 0.08	0.19 ± 0.04	0.31 ± 0.07	0.21 ± 0.02	0.59 ± 0.20	**0.00 ± 0.08**
	Cost ↓	1.81 ± 0.72	5.34 ± 1.33	5.68 ± 0.48	5.57 ± 0.81	1.02 ± 0.22	1.67 ± 1.32	2.10 ± 0.44	5.22 ± 4.02	**0.00 ± 0.00**
AntVelo	Reward ↑	**0.92 ± 0.03**	**-1.01 ± 0.00**	1.00 ± 0.01	**0.97 ± 0.02**	**0.88 ± 0.05**	**0.89 ± 0.01**	**0.95 ± 0.01**	**0.45 ± 0.43**	**0.99 ± 0.01**
	Cost ↓	**0.33 ± 0.19**	**0.00 ± 0.00**	11.82 ± 5.56	**0.59 ± 0.02**	**0.21 ± 0.07**	**0.00 ± 0.00**	**0.76 ± 0.21**	**0.05 ± 0.06**	**0.04 ± 0.02**
HalfCheetahVelo	Reward ↑	**0.86 ± 0.01**	0.67 ± 0.05	**0.63 ± 0.02**	**0.98 ± 0.00**	**0.86 ± 0.01**	**0.89 ± 0.01**	**0.93 ± 0.04**	**0.93 ± 0.02**	**0.94 ± 0.01**
	Cost ↓	**0.30 ± 0.13**	8.56 ± 6.19	**0.00 ± 0.00**	**0.65 ± 0.34**	**0.27 ± 0.06**	**0.00 ± 0.00**	**0.10 ± 0.11**	**0.00 ± 0.00**	**0.00 ± 0.00**
SwimmerVelo	Reward ↑	**0.46 ± 0.07**	0.20 ± 0.20	0.59 ± 0.07	0.66 ± 0.01	0.45 ± 0.06	**-0.02 ± 0.05**	0.60 ± 0.02	**0.05 ± 0.02**	**0.51 ± 0.03**
	Cost ↓	**0.80 ± 0.19**	2.82 ± 4.19	16.80 ± 7.39	1.26 ± 0.50	2.17 ± 2.91	**0.23 ± 0.20**	4.54 ± 3.18	**0.00 ± 0.01**	**0.00 ± 0.00**

D Additional Experimental Results

D.1 Full Main Results

Table 2 is the full version of Table 1, adding the standard deviation across the three training seeds to every entry; safety and best-result marks follow the same convention. Following standard practice in offline safe RL, both tables average 20 evaluation episodes per seed for every method. All other results use 100 evaluation episodes per seed, so the values for RELABEL(COST) can differ slightly between the two.

D.2 Relabel-Rule Ablation: Per-Task Results

Figure 1 summarizes the relabel-rule ablation compactly, as the change in normalized reward relative to our rule with a mark for configurations whose mean cost exceeds the budget. Tables 3 and 4 report the full per-task results behind it: normalized reward, mean episode cost, and safe rate, with 95% bootstrap confidence intervals. All rules share the ReBRAC backbone and hyperparameters and differ only in the reward used for training. The fixed rules relabel the dataset once before training: no relabeling trains on the original rewards and ignores costs; the additive rule uses $r - P_{\mathcal{D}}c$ with the same $P_{\mathcal{D}}$ as ours. The adaptive rules recompute the reward on every minibatch: O3SRL scalarizes

Table 3: Relabel-rule ablation on the Bullet tasks, per task (numbers behind Figure 1). We report normalized reward (R_n), mean episode cost (C), and safe rate (S), the percentage of evaluation episodes with cost within the budget, $C \leq 5$. Bold: configurations whose mean cost is within the budget. Small rows: 95% bootstrap confidence intervals for R_n and C over the 300 evaluation episodes (3 seeds $\times$ 100).

		Fixed rules			Adaptive rules	
Task	**Metric**	**Ours**	**No relabel**	**Additive**	**O3SRL**	**Cost critic**
AntCircle	R_n	**0.83**	1.23	**0.87**	**0.01**	**0.01**
	95% CI	**[0.81,0.85]**	[1.21,1.25]	**[0.85,0.89]**	**[0.01,0.02]**	**[0.01,0.02]**
	C	**0.48**	165.57	**1.53**	**0.00**	**0.00**
	95% CI	**[0.23,0.80]**	[162.19,169.05]	**[0.49,3.34]**	**[0.00,0.00]**	**[0.00,0.00]**
	S (%)	**97**	0	**95**	**100**	**100**
AntRun	R_n	**0.69**	1.07	**0.70**	**0.64**	0.64
	95% CI	**[0.69,0.69]**	[1.06,1.08]	**[0.70,0.70]**	**[0.63,0.65]**	[0.63,0.65]
	C	**2.28**	165.81	**3.11**	**1.53**	8.76
	95% CI	**[2.08,2.49]**	[164.73,166.86]	**[2.92,3.30]**	**[1.34,1.73]**	[8.34,9.18]
	S (%)	**95**	0	**91**	**95**	21
BallCircle	R_n	**0.72**	0.96	**0.74**	**0.73**	**0.49**
	95% CI	**[0.72,0.73]**	[0.96,0.96]	**[0.74,0.74]**	**[0.73,0.73]**	**[0.46,0.53]**
	C	**0.03**	58.55	**0.02**	**0.03**	**0.49**
	95% CI	**[0.01,0.05]**	[58.09,59.03]	**[0.00,0.05]**	**[0.01,0.06]**	**[0.30,0.70]**
	S (%)	**100**	0	**100**	**100**	**95**
BallRun	R_n	**0.27**	1.08	**0.30**	**0.30**	0.27
	95% CI	**[0.27,0.27]**	[1.07,1.08]	**[0.30,0.30]**	**[0.29,0.30]**	[0.27,0.27]
	C	**0.00**	91.00	**0.00**	**0.87**	17.00
	95% CI	**[0.00,0.00]**	[91.00,91.00]	**[0.00,0.00]**	**[0.64,1.13]**	[15.98,18.03]
	S (%)	**100**	0	**100**	**94**	12
CarCircle	R_n	**0.71**	0.87	**0.71**	**0.66**	0.45
	95% CI	**[0.69,0.73]**	[0.84,0.89]	**[0.69,0.72]**	**[0.64,0.68]**	[0.41,0.49]
	C	**0.08**	86.55	**0.28**	**0.00**	22.60
	95% CI	**[0.01,0.17]**	[83.89,88.99]	**[0.12,0.47]**	**[0.00,0.00]**	[15.35,30.50]
	S (%)	**99**	4	**98**	**100**	84
DroneCircle	R_n	**0.58**	0.97	**0.61**	**0.54**	**0.42**
	95% CI	**[0.56,0.60]**	[0.96,0.97]	**[0.60,0.61]**	**[0.54,0.54]**	**[0.39,0.45]**
	C	**0.06**	94.38	**0.02**	**0.00**	**0.17**
	95% CI	**[0.00,0.14]**	[93.68,95.07]	**[0.00,0.04]**	**[0.00,0.00]**	**[0.01,0.43]**
	S (%)	**99**	0	**100**	**100**	**99**
DroneRun	R_n	**0.57**	0.90	**0.58**	**0.57**	**0.56**
	95% CI	**[0.57,0.57]**	[0.89,0.90]	**[0.58,0.58]**	**[0.57,0.57]**	**[0.56,0.56]**
	C	**0.00**	103.80	**0.00**	**4.90**	**0.07**
	95% CI	**[0.00,0.00]**	[102.73,104.93]	**[0.00,0.00]**	**[3.92,5.93]**	**[0.00,0.21]**
	S (%)	**100**	0	**100**	**75**	**100**

it as $r - \lambda\,(c - B/T)$ with a scalar λ chosen by a bandit during training, and the cost-critic rule trains a critic of the current policy's cost and replaces the reward by $-P_{\mathcal{D}}$ wherever the predicted cost-to-go exceeds the per-step share of the budget, $\frac{B}{T}\frac{1-\gamma^T}{1-\gamma}$.

Without relabeling, the policy exceeds the budget on every task, with mean costs between 11.7 and 60 times the budget, which is the source of its higher reward. The additive rule and O3SRL stay within budget on all ten tasks, as does our rule. The additive rule matches our reward to within 0.05 on every task, whereas O3SRL loses substantial reward on AntCircle (0.81) and Swimmer (0.38). The cost-critic rule is the only relabeling rule that violates the budget, on AntRun, BallRun, and CarCircle, and it also loses more than 0.1 reward on five tasks, most on AntCircle (0.81) and Swimmer (0.44).

Table 4: Relabel-rule ablation on the Velocity tasks, per task (numbers behind Figure 1). We report normalized reward (R_n), mean episode cost (C), and safe rate (S), the percentage of evaluation episodes with cost within the budget, $C \leq 10$. Bold: configurations whose mean cost is within the budget. Small rows: 95% bootstrap confidence intervals for R_n and C over the 300 evaluation episodes (3 seeds $\times$ 100).

		Fixed rules			Adaptive rules	
Task	**Metric**	**Ours**	**No relabel**	**Additive**	**O3SRL**	**Cost critic**
Ant	R_n	**0.99**	1.06	**1.00**	**0.97**	**0.95**
	95% CI	**[0.99,0.99]**	[1.06,1.06]	**[1.00,1.00]**	**[0.97,0.97]**	**[0.95,0.95]**
	C	**0.44**	402.20	**1.89**	**0.16**	**0.01**
	95% CI	**[0.37,0.52]**	[397.12,407.34]	**[1.72,2.08]**	**[0.11,0.21]**	**[0.00,0.03]**
	S (%)	**100**	0	**100**	**100**	**100**
HalfCheetah	R_n	**0.94**	1.05	**0.95**	**0.92**	**0.89**
	95% CI	**[0.94,0.94]**	[1.05,1.05]	**[0.95,0.96]**	**[0.92,0.92]**	**[0.89,0.90]**
	C	**0.00**	600.48	**0.00**	**0.00**	**0.00**
	95% CI	**[0.00,0.00]**	[568.25,632.45]	**[0.00,0.01]**	**[0.00,0.00]**	**[0.00,0.00]**
	S (%)	**100**	0	**100**	**100**	**100**
Swimmer	R_n	**0.51**	0.50	**0.50**	**0.13**	**0.07**
	95% CI	**[0.51,0.51]**	[0.50,0.51]	**[0.50,0.51]**	**[0.12,0.15]**	**[0.06,0.07]**
	C	**0.00**	155.84	**0.00**	**0.00**	**0.01**
	95% CI	**[0.00,0.01]**	[154.61,157.09]	**[0.00,0.00]**	**[0.00,0.00]**	**[0.00,0.03]**
	S (%)	**100**	0	**100**	**100**	**100**

D.3 PENALTY-SENSITIVITY DIAGNOSTICS

Table 5 reports the full penalty-scale study for budget $B = 5$. In addition to the default replacement rule ($\lambda = 1$), we evaluate replacement with multipliers $\lambda \in \{0.1, 0.5\}$ and additive penalties with $\lambda \in \{0.5, 1.0\}$. Each configuration uses the same ReBRAC backbone and is evaluated using three independently trained policies with 100 evaluation episodes per policy.

With penalty multiplier λ, the two relabeling rules are

$$\widetilde{r}_\lambda^{\mathrm{rep}} = \begin{cases} r, & c = 0, \\ -\lambda P_{\mathcal{D}}, & c > 0, \end{cases} \qquad \widetilde{r}_\lambda^{\mathrm{add}} = r - \lambda P_{\mathcal{D}}\, c. \tag{D.1}$$

Replacement assigns every cost-bearing transition the same negative reward, whereas the additive rule keeps the original reward and subtracts the penalty, so a transition with $c > 0$ remains positively rewarded whenever $r > \lambda P_{\mathcal{D}} c$.

The results show two distinct forms of sensitivity. First, lowering the replacement penalty eventually degrades safety: at $\lambda = 0.1$, AntRun exceeds the expected-cost budget ($C = 9.12$, with only 10% of episodes within budget), while the remaining six tasks stay below it. Thus the replacement rule still requires a sufficiently strong event charge, consistent with Theorem 1. At $\lambda = 0.5$, replacement keeps the mean cost below the budget on all seven tasks, but the per-episode safe rate already falls on the hardest tasks ($S = 70\%$ on AntRun and 69% on BallRun, versus 95% and 100% for the default). Second, the additive formulation is substantially more sensitive at an intermediate scale. At $\lambda = 0.5$, its mean episode cost exceeds $B = 5$ on AntRun (8.82), BallRun (76.03), and DroneCircle (84.36), while replacement at the same multiplier remains below the expected-cost budget on all seven tasks. On BallRun and DroneCircle the additive policy attains higher normalized reward than any safe configuration ($R_n = 1.05$ and 0.92) while almost never staying within budget ($S = 8\%$ and 0%): the relabeled data still reward the costly behavior. At $\lambda = 1.0$ the additive rule is safe on every task (mean cost at most 3.11).

To examine the mechanism suggested by Equation D.1, define

$$q_\lambda = \Pr_{\mathcal{D}}\left(r > \lambda P_{\mathcal{D}} \mid c > 0\right). \tag{D.2}$$

The quantity q_λ is the fraction of observed positive-cost transitions that remain positively rewarded under additive relabeling (costs are binary on these tasks, so $r - \lambda P_{\mathcal{D}} c > 0$ reduces to $r > \lambda P_{\mathcal{D}}$).

Table 5: Relabel(Cost) penalty scale and relabeling mode, on the Bullet tasks. We report normalized reward (R_n), mean episode cost (C), and safe rate (S), the percentage of evaluation episodes with cost within the budget, $C \leq 5$. Bold: configurations whose mean cost is within the budget. Small rows: 95% bootstrap confidence intervals for R_n and C over the 300 evaluation episodes (3 seeds $\times$ 100). λ scales the penalty P (Base: replace with $\lambda = 1$).

		Replace: $r \to -P$			**Additive:** $r \to r - Pc$	
Task	**Metric**	**Base**	$\boldsymbol{\lambda = 0.1}$	$\boldsymbol{\lambda = 0.5}$	$\boldsymbol{\lambda = 0.5}$	$\boldsymbol{\lambda = 1.0}$
AntCircle	R_n	**0.83**	**0.66**	**0.91**	**0.66**	**0.87**
	95% CI	**[0.81,0.85]**	**[0.61,0.71]**	**[0.89,0.93]**	**[0.61,0.71]**	**[0.85,0.89]**
	C	**0.48**	**1.20**	**0.76**	**3.86**	**1.53**
	95% CI	**[0.23,0.80]**	**[0.87,1.56]**	**[0.52,1.03]**	**[1.79,7.03]**	**[0.50,3.39]**
	S (%)	**97**	**91**	**96**	**85**	**95**
AntRun	R_n	**0.69**	0.71	**0.70**	0.71	**0.70**
	95% CI	**[0.69,0.69]**	[0.71,0.71]	**[0.70,0.70]**	[0.71,0.71]	**[0.70,0.70]**
	C	**2.28**	9.12	**4.58**	8.82	**3.11**
	95% CI	**[2.08,2.49]**	[8.79,9.45]	**[4.35,4.81]**	[8.48,9.16]	**[2.91,3.30]**
	S (%)	**95**	10	**70**	15	**91**
BallCircle	R_n	**0.72**	**0.76**	**0.75**	**0.75**	**0.74**
	95% CI	**[0.72,0.73]**	**[0.76,0.76]**	**[0.74,0.75]**	**[0.74,0.75]**	**[0.74,0.74]**
	C	**0.03**	**0.76**	**0.04**	**0.71**	**0.02**
	95% CI	**[0.01,0.05]**	**[0.62,0.90]**	**[0.02,0.08]**	**[0.57,0.85]**	**[0.00,0.05]**
	S (%)	**100**	**100**	**100**	**99**	**100**
BallRun	R_n	**0.27**	**0.30**	**0.30**	1.05	**0.30**
	95% CI	**[0.27,0.27]**	**[0.30,0.30]**	**[0.30,0.30]**	[1.01,1.08]	**[0.30,0.30]**
	C	**0.00**	**2.65**	**3.63**	76.03	**0.00**
	95% CI	**[0.00,0.00]**	**[2.42,2.87]**	**[3.00,4.28]**	[72.84,79.07]	**[0.00,0.00]**
	S (%)	**100**	**96**	**69**	8	**100**
CarCircle	R_n	**0.71**	**0.74**	**0.74**	**0.74**	**0.71**
	95% CI	**[0.69,0.73]**	**[0.72,0.75]**	**[0.72,0.75]**	**[0.73,0.76]**	**[0.69,0.72]**
	C	**0.08**	**2.05**	**0.30**	**4.60**	**0.28**
	95% CI	**[0.01,0.17]**	**[1.50,2.64]**	**[0.13,0.50]**	**[3.75,5.52]**	**[0.12,0.47]**
	S (%)	**99**	**88**	**97**	**82**	**98**
DroneCircle	R_n	**0.58**	**0.61**	**0.61**	0.92	**0.61**
	95% CI	**[0.56,0.60]**	**[0.60,0.61]**	**[0.61,0.61]**	[0.92,0.93]	**[0.60,0.61]**
	C	**0.06**	**0.13**	**0.02**	84.36	**0.02**
	95% CI	**[0.00,0.14]**	**[0.02,0.30]**	**[0.00,0.05]**	[83.88,84.85]	**[0.00,0.04]**
	S (%)	**99**	**99**	**100**	0	**100**
DroneRun	R_n	**0.57**	**0.59**	**0.58**	**0.59**	**0.58**
	95% CI	**[0.57,0.57]**	**[0.59,0.59]**	**[0.58,0.58]**	**[0.59,0.60]**	**[0.58,0.58]**
	C	**0.00**	**0.00**	**0.30**	**0.00**	**0.00**
	95% CI	**[0.00,0.00]**	**[0.00,0.00]**	**[0.00,0.70]**	**[0.00,0.00]**	**[0.00,0.00]**
	S (%)	**100**	**100**	**99**	**100**	**100**

Replacement has no analogous failure mode because every positive-cost transition receives $-\lambda P_{\mathcal{D}} < 0$. Table 6 reports q_λ on each training dataset. At $\lambda = 1$, $q_1 = 0$ on every task by construction, since $P_{\mathcal{D}} = \max_i r_i$; this matches the additive rule being safe at that scale. At $\lambda = 0.5$, between 9% and 85% of positive-cost transitions keep a positive reward, and the three tasks on which additive relabeling violates the budget all have $q_{0.5} \geq 37\%$. A large q_λ is, however, not sufficient for failure: CarCircle ($q_{0.5} = 75.5\%$) and BallCircle (48.7%) remain within budget, although CarCircle only narrowly ($C = 4.60$, $S = 82\%$). We therefore read q_λ as identifying when the additive rule leaves costly behavior rewarded, not as a predictor of the resulting cost, which also depends on how often the learned policy can reach those transitions.

The confidence intervals in Table 5 are obtained by pooling the evaluation episodes from the three trained policies and bootstrapping over episodes. They therefore characterize rollout variability for the reported trained policies rather than uncertainty over retraining the algorithm.

Table 6: Share q_λ of positive-cost transitions that remain positively rewarded under additive relabeling (Eq. D.2), computed on the training datasets, with $P_\mathcal{D} = \max_{i \in \mathcal{D}} r_i$.

	AntCircle	AntRun	BallCircle	BallRun	CarCircle	DroneCircle	DroneRun
$P_\mathcal{D}$	2.37	7.97	8.83	21.99	2.61	4.64	7.19
$q_{0.5}$ (%)	9.1	70.7	48.7	37.0	75.5	84.8	22.5
$q_{1.0}$ (%)	0.0	0.0	0.0	0.0	0.0	0.0	0.0

D.4 Filtering and dataset composition

The filtering experiment removes trajectories whose cumulative cost is at most a specified threshold B_f and trains RELABEL(COST) using only the remaining trajectories. Thus every trajectory retained under threshold B_f satisfies $C_\tau > B_f$. The transition-level relabeling rule itself is unchanged; the penalty $P_\mathcal{D}$ is recomputed on the retained data. As a transition-level counterpart (Drop), we instead remove every transition with $c > 0$ and keep all zero-cost transitions. Results are reported in Table 7.

Table 8 reports the amount of data retained after filtering. Low-cost trajectories are rare in these datasets: fewer than 10% of trajectories have $C_\tau \leq 5$ on six of the seven tasks (24% on DroneRun), so $B_f = 5$ still keeps 76–96% of the transitions. Increasing B_f removes progressively more trajectories; at $B_f = 80$, 25–57% of the transitions remain, and nothing remains on the Ball tasks, whose costliest trajectories have $C_\tau = 80$. Table 9 shows that zero-cost transitions remain the majority of the retained data on six of the seven tasks up to $B_f = 40$ (51–82%), even though every retained trajectory violates the filtering threshold. The exception is BallRun, whose dataset is only 51% zero-cost before filtering and 37–49% after it.

This follows from the difference between trajectory cost and transition cost. With binary per-step costs,

$$C_\tau = \sum_{t=0}^{T_\tau - 1} c_t, \qquad c_t \in \{0, 1\}, \tag{D.3}$$

where T_τ is the trajectory length. A trajectory can therefore satisfy $C_\tau > B_f$ while having $c_t = 0$ on most of its transitions. Those zero-cost transitions retain their original reward under RELABEL(COST), while positive-cost transitions receive the replacement penalty.

The evaluation results remain within budget on all tasks up to $B_f = 10$, with the exception of a lower safe rate on AntRun at $B_f = 5$ ($S = 73\%$, mean cost 3.93). AntRun is also the first task to exceed the budget, at $B_f = 20$ ($C = 6.20$); at $B_f = 40$ it is joined by BallRun ($C = 9.65$), and at $B_f = 80$ every task on which the filter leaves data exceeds the budget (C between 11.1 and 42.1). We examined whether AntRun's earlier degradation could be explained by unusually severe data removal or by longer runs of consecutive cost events in the retained data. Neither distinguishes it: at $B_f = 20$ AntRun retains 77.9% of its transitions, within the 65–87% range of the other tasks, and its cost-event runs are the shortest (7.5 steps on average, versus 9.8–58.8 on the other tasks). We therefore report the behavior without assigning a specific cause.

Drop isolates the role of the positive-cost transitions. It keeps 51–83% of the transitions, comparable to moderate trajectory filtering, but none with $c > 0$, so the relabeling has no costly transition to penalize. The resulting policies exceed the budget on every task (mean cost 13.2–95.1, safe rate at most 15%) while exceeding the reward of the default on every task. Trajectory filtering keeps the costly transitions and degrades far more gradually. RELABEL(COST) therefore depends on observing positive-cost transitions, not on observing trajectories that satisfy the budget.

The same statistics characterize the datasets before any filtering. Complete low-cost trajectories are uncommon: fewer than 10% of trajectories have $C_\tau \leq 5$ on six of the seven tasks (24% on DroneRun), as implied by the $B_f = 5$ column of Table 8. Individual zero-cost transitions, by contrast, remain abundant, making up 51–83% of each dataset (Table 9, column All). This separation between trajectory-level feasibility and transition-level coverage is the empirical setting in which local cost relabeling is most useful.

Table 7: Relabel(Cost) with degraded training data, on the Bullet tasks. We report normalized reward (R_n), mean episode cost (C), and safe rate (S), the percentage of evaluation episodes with cost within the budget, $C \leq 5$. Bold: configurations whose mean cost is within the budget. Small rows: 95% bootstrap confidence intervals for R_n and C over the 300 evaluation episodes (3 seeds $\times$ 100). FB b: only trajectories with cost $> b$ are kept for training; Drop: all transitions with cost > 0 are removed. –: not run (FB 80 has no Ball tasks).

			Replace $r \to -P$, degraded training data					
Task	**Metric**	**Base**	**FB 5**	**FB 10**	**FB 20**	**FB 40**	**FB 80**	**Drop**
AntCircle	R_n	**0.83**	**0.81**	**0.83**	**0.80**	**0.79**	0.56	0.97
	95% CI	**[0.81,0.85]**	**[0.78,0.83]**	**[0.81,0.85]**	**[0.77,0.82]**	**[0.78,0.81]**	[0.53,0.60]	[0.95,0.99]
	C	**0.48**	**2.89**	**0.81**	**0.62**	**0.79**	11.12	22.57
	95% CI	**[0.23,0.80]**	**[0.83,5.69]**	**[0.34,1.50]**	**[0.33,0.97]**	**[0.51,1.10]**	[6.71,16.50]	[19.43,26.58]
	S (%)	**97**	**93**	**96**	**96**	**94**	68	15
AntRun	R_n	**0.69**	**0.68**	**0.65**	0.64	0.56	0.50	0.73
	95% CI	**[0.69,0.69]**	**[0.68,0.69]**	**[0.65,0.65]**	[0.64,0.64]	[0.55,0.57]	[0.49,0.50]	[0.72,0.73]
	C	**2.28**	**3.93**	**2.98**	6.20	5.12	11.71	27.65
	95% CI	**[2.08,2.49]**	**[3.64,4.21]**	**[2.79,3.18]**	[5.91,6.48]	[4.78,5.45]	[10.64,12.87]	[26.41,28.91]
	S (%)	**95**	**73**	**92**	40	54	28	0
BallCircle	R_n	**0.72**	**0.73**	**0.72**	**0.70**	**0.62**	–	0.77
	95% CI	**[0.72,0.73]**	**[0.73,0.73]**	**[0.71,0.73]**	**[0.70,0.71]**	**[0.60,0.63]**	–	[0.76,0.77]
	C	**0.03**	**0.23**	**0.01**	**0.06**	**2.23**	–	13.22
	95% CI	**[0.01,0.06]**	**[0.15,0.32]**	**[0.00,0.03]**	**[0.03,0.10]**	**[1.85,2.61]**	–	[12.63,13.84]
	S (%)	**100**	**100**	**100**	**100**	**77**	–	4
BallRun	R_n	**0.27**	**0.30**	**0.30**	**0.28**	0.28	–	0.34
	95% CI	**[0.27,0.27]**	**[0.30,0.30]**	**[0.30,0.30]**	**[0.28,0.28]**	[0.28,0.29]	–	[0.34,0.34]
	C	**0.00**	**0.00**	**0.02**	**0.79**	9.65	–	54.47
	95% CI	**[0.00,0.00]**	**[0.00,0.00]**	**[0.01,0.04]**	**[0.67,0.91]**	[8.52,10.79]	–	[53.57,55.38]
	S (%)	**100**	**100**	**100**	**100**	51	–	0
CarCircle	R_n	**0.71**	**0.70**	**0.72**	**0.72**	**0.69**	0.56	0.72
	95% CI	**[0.69,0.72]**	**[0.68,0.72]**	**[0.71,0.74]**	**[0.70,0.73]**	**[0.67,0.70]**	[0.54,0.59]	[0.70,0.74]
	C	**0.08**	**0.02**	**0.06**	**0.02**	**0.57**	42.13	29.68
	95% CI	**[0.01,0.18]**	**[0.00,0.07]**	**[0.00,0.16]**	**[0.00,0.05]**	**[0.29,0.91]**	[37.47,47.39]	[27.82,31.50]
	S (%)	**99**	**100**	**99**	**100**	**97**	6	11
DroneCircle	R_n	**0.58**	**0.55**	**0.49**	**0.49**	**0.43**	0.08	0.60
	95% CI	**[0.56,0.60]**	**[0.52,0.57]**	**[0.45,0.52]**	**[0.47,0.51]**	**[0.40,0.45]**	[0.05,0.10]	[0.58,0.62]
	C	**0.06**	**0.59**	**0.67**	**1.74**	**0.96**	22.13	13.49
	95% CI	**[0.00,0.14]**	**[0.33,0.90]**	**[0.29,1.15]**	**[1.10,2.44]**	**[0.50,1.50]**	[19.42,25.22]	[13.02,13.96]
	S (%)	**99**	**95**	**97**	**90**	**94**	20	5
DroneRun	R_n	**0.57**	**0.56**	**0.56**	**0.54**	**0.55**	0.51	0.65
	95% CI	**[0.57,0.57]**	**[0.56,0.56]**	**[0.56,0.56]**	**[0.54,0.54]**	**[0.55,0.55]**	[0.50,0.53]	[0.65,0.66]
	C	**0.00**	**0.09**	**0.00**	**0.04**	**1.51**	29.95	95.12
	95% CI	**[0.00,0.00]**	**[0.00,0.28]**	**[0.00,0.00]**	**[0.00,0.11]**	**[1.13,1.92]**	[27.40,32.62]	[93.92,96.29]
	S (%)	**100**	**100**	**100**	**100**	**86**	18	0

D.5 Robustness to training seeds

All results in the paper use three training seeds (10, 20, 30). To check that our conclusions do not depend on this choice, we trained our method with three additional seeds (40, 50, 60) on every Bullet-Safety-Gym task and on the Ant, HalfCheetah, and Swimmer velocity tasks, with the same hyperparameters and evaluation protocol. Table 10 compares the three-seed results with the six-seed results. For six seeds, the confidence interval is taken over the six per-seed means rather than over evaluation episodes, so it estimates how much the result would change if training were rerun with a new seed.

Safety is unchanged with six seeds: the mean cost remains within the budget on every task, and so does the mean cost of every individual seed (at most 3.83, on AntRun, against a budget of 5). The

Table 8: Training data left by the filters of Table 7. FB B (`filter_budget`): trajectories with total cost $\leq B$ are removed; entries give the percentage of the dataset's transitions / trajectories retained. Drop: transitions with cost > 0 are removed; entries give the percentage of transitions retained (every trajectory keeps its cost-0 steps, so none is removed whole). †No trajectory has cost > 80 on the Ball tasks, so FB 80 would remove all data and was not run.

	Original dataset		FB: transitions / trajectories retained (%)					Drop (%)
Task	Traj.	Transitions	$B = 5$	$B = 10$	$B = 20$	$B = 40$	$B = 80$	trans.
AntCircle	5,728	2,510,820	95.9 / 94.0	93.1 / 91.0	87.2 / 84.5	76.2 / 71.9	57.2 / 51.6	80.6
AntRun	1,816	363,010	90.8 / 90.8	86.1 / 86.1	77.9 / 77.9	64.0 / 64.0	41.3 / 41.3	66.0
BallCircle	886	177,200	94.5 / 94.5	90.3 / 90.3	79.9 / 79.9	56.5 / 56.5	0.0 / 0.0†	78.4
BallRun	940	94,000	95.1 / 95.1	91.0 / 91.0	83.7 / 83.7	66.5 / 66.5	0.0 / 0.0†	51.3
CarCircle	1,450	435,000	93.2 / 93.2	88.6 / 88.6	78.0 / 78.0	56.3 / 56.3	24.5 / 24.5	83.4
DroneCircle	1,923	576,243	90.9 / 90.9	85.8 / 85.8	77.8 / 77.8	62.2 / 62.1	33.2 / 33.1	82.1
DroneRun	1,990	395,099	75.8 / 76.0	71.8 / 72.0	65.1 / 65.2	53.0 / 53.1	29.1 / 28.9	74.6

Table 9: Composition of the data retained by trajectory filtering (Table 8). Top: share of retained transitions with zero cost (%). Bottom: mean length, in steps, of runs of consecutive positive-cost transitions within retained trajectories. All: unfiltered dataset; –: the filter removes all data.

Task	All	$B_f = 5$	$B_f = 10$	$B_f = 20$	$B_f = 40$	$B_f = 80$
Zero-cost share of retained transitions (%)						
AntCircle	81	80	79	78	76	73
AntRun	66	63	61	57	51	42
BallCircle	78	77	76	74	70	–
BallRun	51	49	47	44	37	–
CarCircle	83	82	81	80	76	71
DroneCircle	82	80	79	78	75	70
DroneRun	75	67	65	62	57	47
Mean length of consecutive cost-event runs (steps)						
AntCircle	34.5	34.7	35.1	35.9	37.5	39.8
AntRun	6.7	6.8	6.9	7.5	8.9	12.8
BallCircle	9.5	9.5	9.6	9.8	10.3	–
BallRun	34.8	35.7	36.8	38.6	42.9	–
CarCircle	21.5	21.6	22.0	22.5	23.1	23.9
DroneCircle	12.1	12.2	12.3	12.5	12.9	13.7
DroneRun	51.8	52.8	55.0	58.8	65.5	80.1

normalized reward changes by at most 0.01 on ten of the eleven tasks. The exception is DroneCircle, where one of the additional seeds (seed 50) reaches only $R_n = -0.22$ while remaining safe; this single run lowers the six-seed mean from 0.58 to 0.46 and widens the across-seed interval to $[0.18, 0.60]$, while the other five seeds reach between 0.54 and 0.61.

E Implementation and Experimental Details

E.1 ReBRAC implementation

We use ReBRAC (Tarasov et al., 2023a) as the offline RL backbone. The reward transformation is applied once before training. Given the offline dataset $\mathcal{D}$, we compute

$$P_{\mathcal{D}} = \max_{i \in \mathcal{D}} |r_i|, \tag{E.1}$$

using a predetermined fallback $P_{\min} > 0$ if $P_{\mathcal{D}} = 0$, and set

$$\widetilde{r}_i = \begin{cases} r_i, & c_i = 0, \\ -P_{\mathcal{D}}, & c_i > 0. \end{cases} \tag{E.2}$$

Table 10: Robustness to training seeds: our method with the three training seeds used throughout the paper (10, 20, 30) and with three additional seeds (40, 50, 60), 100 evaluation episodes per seed. R_n: normalized reward; C: mean episode cost; S: percentage of episodes within the budget. Second line: 95% bootstrap confidence interval, over evaluation episodes for three seeds (as in the other tables) and over the six per-seed means for six seeds. Bold: mean cost within the budget.

	3 seeds			6 seeds		
Task	R_n	C	S (%)	R_n	C	S (%)
Bullet-Safety-Gym (budget 5)						
AntCircle	**0.83** [0.81,0.85]	**0.48** [0.23,0.80]	**97**	**0.83** [0.82,0.83]	**1.16** [0.55,2.09]	**95**
AntRun	**0.69** [0.69,0.69]	**2.28** [2.08,2.49]	**95**	**0.69** [0.69,0.69]	**1.75** [1.24,2.62]	**97**
BallCircle	**0.72** [0.72,0.73]	**0.03** [0.01,0.05]	**100**	**0.73** [0.72,0.74]	**0.05** [0.02,0.09]	**100**
BallRun	**0.27** [0.27,0.27]	**0.00** [0.00,0.00]	**100**	**0.28** [0.26,0.30]	**0.00** [0.00,0.00]	**100**
CarCircle	**0.71** [0.69,0.72]	**0.08** [0.01,0.18]	**99**	**0.70** [0.68,0.72]	**0.04** [0.01,0.09]	**100**
CarRun	**0.90** [0.90,0.90]	**0.58** [0.47,0.69]	**100**	**0.90** [0.89,0.91]	**0.92** [0.46,1.41]	**97**
DroneCircle	**0.58** [0.56,0.60]	**0.06** [0.00,0.14]	**99**	**0.46** [0.18,0.60]	**0.30** [0.01,0.83]	**98**
DroneRun	**0.57** [0.57,0.57]	**0.00** [0.00,0.00]	**100**	**0.57** [0.56,0.58]	**0.00** [0.00,0.00]	**100**
Safety-Gymnasium velocity (budget 10)						
AntVelocity	**0.99** [0.99,0.99]	**0.44** [0.37,0.53]	**100**	**0.99** [0.98,0.99]	**0.45** [0.33,0.58]	**100**
HalfCheetahVelocity	**0.94** [0.94,0.94]	**0.00** [0.00,0.00]	**100**	**0.94** [0.94,0.94]	**0.00** [0.00,0.00]	**100**
SwimmerVelocity	**0.51** [0.51,0.51]	**0.00** [0.00,0.01]	**100**	**0.51** [0.49,0.53]	**0.02** [0.00,0.05]	**100**

All transitions are retained. ReBRAC is then trained on the transformed dataset with its standard actor and critic regularization.

For a minibatch $\mathcal{B}$, ReBRAC normalizes the critic contribution to the actor objective using

$$\lambda_Q = \mathrm{sg}\left(\frac{1}{\frac{1}{|\mathcal{B}|}\sum_{s\in\mathcal{B}}\left|\min_j Q_{\phi_j}(s,\pi_\theta(s))\right|}\right), \tag{E.3}$$

where sg denotes stop-gradient. With configured actor regularization coefficient β_π, the actor loss is

$$\mathcal{L}_\pi(\theta) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[-\lambda_Q \min_j Q_{\phi_j}(s,\pi_\theta(s)) + \beta_\pi \left\|\pi_\theta(s)-a\right\|^2\right]. \tag{E.4}$$

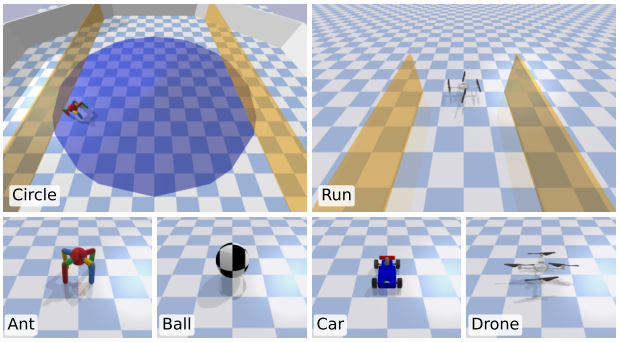
Dividing by the positive factor λ_Q gives the corresponding regularization coefficient in value units,

$$\beta_{\text{eff}} = \frac{\beta_\pi}{\lambda_Q}. \tag{E.5}$$

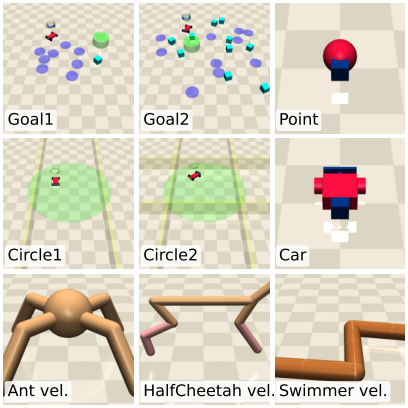
This is the coefficient that corresponds to β in Theorem 4. Since λ_Q changes with the critic and minibatch, β_{eff} can vary during training even though β_π is fixed.

The critic is trained on the transformed reward using ReBRAC's continuation regularizer. Writing a' for the target actor action after TD3 target smoothing and $a'_{\mathcal{D}}$ for the recorded successor action,

$$\begin{aligned} y &= \widetilde{r} + \gamma(1-d)\left[\min_j Q_{\bar{\phi}_j}(s',a') - \beta_Q \left\|a'-a'_{\mathcal{D}}\right\|^2\right], \\ \mathcal{L}_{Q_j} &= \mathbb{E}_{\mathcal{D}}\left[\left(Q_{\phi_j}(s,a)-\mathrm{sg}(y)\right)^2\right]. \end{aligned} \tag{E.6}$$



(a) Bullet-Safety-Gym: Circle and Run tasks; Ant, Ball, Car, and Drone agents.



(b) Safety-Gymnasium: Goal and Circle tasks, Point and Car agents, velocity tasks.

Figure 3: Visualizations of benchmark environments. Images from Gronauer (2022) and Ji et al. (2023).

The configured coefficients β_π and β_Q are fixed during each run. The theory in Section 4 gives a sufficient condition for actor-side anchoring; it does not require or analyze the critic regularizer separately.

We build on the ReBRAC implementation of Unifloral (Jackson et al., 2025) (`https://github.com/EmptyJackson/unifloral`) and use its configuration (`configs/unifloral/rebrac.yaml`): two critics, layer normalization in the actor and critic, a deterministic $\tanh$ actor, two critic updates per actor update, TD3 target smoothing with noise 0.2 clipped to 0.5, learning rate 10^{-3}, batch size 1024, $\gamma = 0.99$, and Polyak step 0.005; we additionally standardize observations. Rather than sweeping them, we choose β_π per task from $\{0.001, 0.01, 0.1\}$, three of the values that the ReBRAC authors' reference implementation (CORL (Tarasov et al., 2023b), `https://github.com/tinkoff-ai/CORL`) uses in its per-dataset D4RL configurations: $\beta_\pi = 0.001$ on AntCircle; 0.1 on AntRun, DroneRun, CarCircle1, CarCircle2, and the velocity tasks; and 0.01 on all other tasks. We fix the critic coefficient to $\beta_Q = 0.01$ on all tasks without tuning; it is also the critic value that appears most often in those configurations.

E.2 Environment Descriptions

Our experiments span two safety-oriented continuous-control benchmarks implemented on different physics simulators. The selected tasks expose agents to several forms of constraints, including workspace boundaries, hazard avoidance, and limits on velocity.

Bullet-Safety-Gym (Gronauer, 2022): Bullet-Safety-Gym is implemented in PyBullet and provides multiple embodied agents, including Ball, Car, Drone, and Ant. We use the Run and Circle task families. Run evaluates whether an agent can make progress through a corridor without violating spatial or speed-related constraints. The corridor limits are virtual rather than physical barriers, so an agent may cross them, but doing so incurs a safety cost. Additional cost is assigned when the agent exceeds the allowed speed.

Circle instead requires sustained clockwise locomotion around a circular region. The reward encourages fast motion along the circle, whereas crossing either of two boundary lines placed on the sides of the circle incurs a safety cost; as in Run, the boundaries are virtual. Example scenes and the four agents are shown in Figure 3(a).

Safety-Gymnasium (Ray et al., 2019; Ji et al., 2023): Safety-Gymnasium is built on MuJoCo and contains a broader collection of constrained-control problems. For the Point and Car morphologies, we consider Goal and Circle tasks at two difficulty levels (1 and 2). The Goal setting requires navigation toward a target while avoiding hazardous regions; level 2 contains more hazards and additional movable obstacles. Once a target is reached, the environment assigns a new goal location. In the

Circle setting, the agent is expected to move along a designated green circular region; crossing the surrounding boundaries generates cost, and level 2 adds boundaries along a second axis, restricting the admissible area further. Figure 3(b) illustrates representative instances.

We also use velocity-constrained locomotion tasks involving Ant, HalfCheetah, and Swimmer. Unlike the navigation tasks, these environments focus on forward movement generated through coordinated body motion. Safety is enforced through a velocity limit, so the agent must balance locomotion performance against violations of the prescribed speed constraint.