REACHABILITY-AWARE DIFFUSION POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion policies provide expressive action distributions for continuous-control reinforcement learning. However, safety-aware online diffusion policy optimization remains underexplored, particularly methods that use predictive reachability information without an explicit dynamics model. We propose Reachability-Aware Diffusion Policy Optimization (RADPO), a model-free method that combines predictive first-hit safety estimation with cumulative-cost budget feedback. RADPO learns a discounted first-hit reachability value that captures the discounted risk of a cost event, assigns larger weight to events that occur sooner, and uses this signal to shape the reward. A separate dual-like multiplier adjusts the shaping strength according to realized episodic costs relative to a prescribed budget. The diffusion actor improves through weighted denoising regression on candidate actions scored by the reward critic. Our approach requires neither a learned dynamics model, action gradients through the critics, nor differentiation through the reverse diffusion sampler. We establish theoretical properties of the reachability value and show that accumulated reachability penalty provides a conservative surrogate for future discounted cumulative cost. Across ten continuous-control safety tasks, RADPO achieves competitive reward-cost trade-offs, with substantial reductions in constraint violations on several tasks relative to the compared baselines. Our theoretical and empirical analysis supports that combining reachability with cumulative budget feedback is a viable approach to safety-aware diffusion policies.

1 INTRODUCTION

A common formulation of safe reinforcement learning aims to maximize task performance while keeping expected cumulative safety cost below some budget (Achiam et al., 2017; Stooke et al., 2020). A central challenge is anticipating unsafe outcomes: an action may incur no immediate cost yet move the agent toward future violations. Cost critics address this by estimating accumulated future cost (Liu et al., 2022; Wu et al., 2024), which captures violation magnitude and repeated exposure but does not explicitly isolate when the first cost encounter occurs. Reachability-based safety estimation provides a complementary perspective by explicitly predicting future constraint violations (Yu et al., 2022; Ganai et al., 2023) instead. Discounted first-hit reachability captures whether a cost will be encountered while assigning greater weight to earlier encounters. We investigate whether this predictive signal can guide policy improvement while cumulative-cost feedback separately regulates the safety budget.

Diffusion policies provide expressive action distributions for continuous control (Wang et al., 2023; Chi et al., 2024) and can be optimized online through value-weighted denoising regression (Ma et al., 2025). However, most online diffusion-policy optimization methods focus on reward maximization (Yang et al., 2023; Psenka et al., 2024; Ren et al., 2024; Ma et al., 2025), while incorporating predictive safety information without an explicit dynamics model is relatively underexplored. We therefore propose **Reachability-Aware Diffusion Policy Optimization (RADPO)**, a model-free approach that combines discounted first-hit reachability with cumulative-cost budget feedback.

RADPO learns a discounted first-hit reachability value and uses the predicted reachability of successor states to shape the reward. An adaptive safety multiplier controls the shaping strength according to realized episodic costs relative to the prescribed budget, preserving information about cost magnitudes and repeated violations. The diffusion actor then improves through weighted denoising

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.

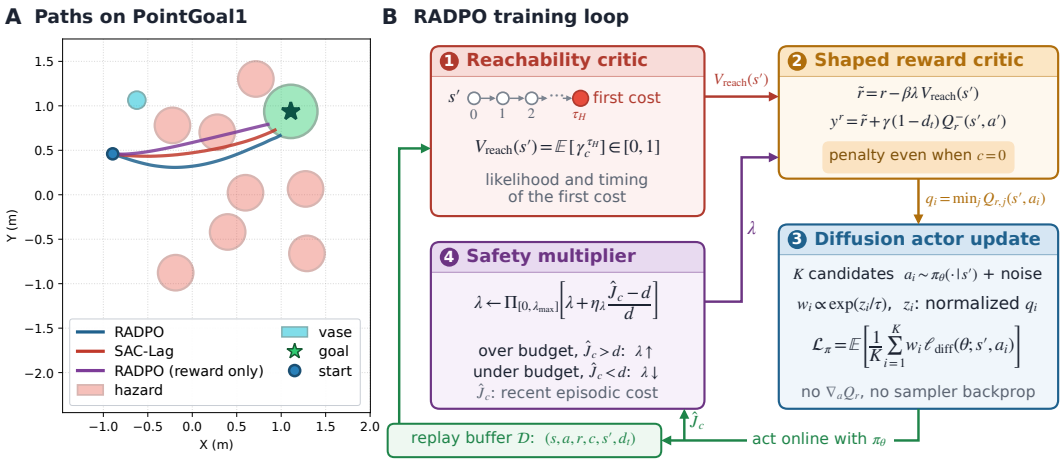

Figure 1: **(A)** Evaluation paths obtained from fully trained policies on the same fixed PointGoal1 layout. See Figure 6 for clearance distribution of these paths. **(B)** Illustrative diagram for the RADPO algorithm.

regression on candidate actions scored by the reward critic. This requires neither a learned dynamics model, action gradients through the critics, nor differentiation through the reverse diffusion sampler. This mechanism guides learning toward safer behavior. Figure 1.A illustrates this behavior in a navigation scenario. SAC-Lag follows a greedier path close to the hazard boundaries and occasionally enters them, whereas RADPO takes a more conservative route with greater clearance.

Our main contributions are:

- We introduce RADPO, a novel algorithm that combines discounted first-hit reachability with cumulative-cost budget feedback and integrates these signals into online diffusion policy optimization through value-weighted denoising regression.
- We provide theoretical results for the discounted reachability value, including its first-hit interpretation and bounds relating accumulated reachability to discounted cumulative cost and repeated cost exposure. We further establish finite-candidate mirror-descent guarantees and analyze the safety-multiplier controller's response to observed budget violations.
- We evaluate RADPO on ten continuous-control safety tasks, showing competitive reward–cost trade-offs and substantial reductions in constraint violations on several tasks relative to the compared baselines, complemented by design ablations.

2 Related Work

Constrained reinforcement learning. Safe reinforcement learning is commonly formulated using constrained Markov decision processes (CMDPs), where expected return is maximized subject to cumulative-cost constraints. Some methods (Achiam et al., 2017; Yang et al., 2020; Zhang et al., 2020; Liu et al., 2022; Milosevic et al., 2025) enforce these constraints through constrained or trust-region policy updates, while primal-dual approaches (Tessler et al., 2018; Stooke et al., 2020; Wu et al., 2024) adapt Lagrange multipliers according to observed constraint violations. Beyond changing the policy update itself, other works modify the learned safety signal used by the optimizer; for example, USPC (Demiray et al., 2026) replaces standard cost estimates with uncertainty-aware conservative safety surrogates. RADPO follows a similar approach, but instead introduces a distinct predictive reachability value alongside the cumulative-cost signal: episodic costs adapt the safety multiplier, while reachability shapes the reward used for policy improvement.

Reachability-based safety and control. Reachability analysis attempts to determine whether target or unsafe sets can be reached under a dynamical system. Classical Hamilton-Jacobi (HJ) methods compute backward reachable or viable sets from known dynamics (Mitchell et al., 2005; Bansal et al., 2017), with subsequent work connecting reachability analysis to learning-based control and reinforcement learning (Fisac et al., 2018; Akametalu et al., 2024; Li et al., 2025). More recently,

λ-Reachability (Chen et al., 2026) introduces geometric-horizon Bellman targets that interpolate between discounted local updates and long-horizon max-over-trajectory safety values. In model-free safe RL, RCRL (Yu et al., 2022) learns a reachability-based safety value to characterize feasible states, while RESPO (Ganai et al., 2023) extends this idea to stochastic systems using a learned estimate of future constraint violation. RADPO instead targets a policy-conditional discounted first-hit cost value, assigning greater weight to violations occurring sooner. This predictive signal shapes the reward, while cumulative episodic cost is handled separately through budget feedback.

Online diffusion policy optimization. Several recent methods optimize diffusion policies using online reinforcement learning. DIPO (Yang et al., 2023) and Q-Score Matching (Psenka et al., 2024) use critic gradients to improve actions or policy scores, respectively. Other approaches optimize a Q-weighted variational objective (Ding et al., 2024), backpropagate critic gradients through the diffusion sampling process (Wang et al., 2024), or apply PPO-style policy-gradient optimization to the denoising process (Ren et al., 2024; Zou et al., 2026; Liu et al., 2026). Most closely related to RADPO, DPMD (Ma et al., 2025) derives diffusion-policy improvement from policy mirror descent using value-weighted denoising regression, avoiding both critic action gradients and differentiation through the reverse sampler. RADPO builds on this policy-improvement mechanism but evaluates candidate actions using a reward critic learned from rewards shaped by predictive reachability and an adaptive safety multiplier.

Constrained and safety-aware diffusion policies. Safety and constraints have also been incorporated into diffusion-based decision making. Offline methods use cost-conditioned or feasibility-guided trajectory generation (Lin et al., 2024; Zheng et al., 2024), while barrier- and constraint-guided planners incorporate safety directly into diffusion sampling or denoising (Xiao et al., 2025; Mizuta & Leung, 2024; Zhang et al., 2025). In the online setting, ALGD (Cheng et al., 2026) uses an augmented-Lagrangian energy to guide diffusion-policy optimization under safety costs. Beyond environmental safety, diffusion policies have also been subject to behavior- or trust-region constraints that limit deviation during offline or online optimization (Gao et al., 2025; Le et al., 2026; Simsir & Oguz, 2026). RADPO instead combines fully online model-free diffusion-policy learning with a policy-conditional first-hit signal, separate episodic budget feedback, and value-weighted denoising without critic gradients or reverse-sampler differentiation.

3 PRELIMINARIES

Constrained Markov Decision Processes We consider a CMDP with states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, transition kernel P, reward r, nonnegative cost c, and initial-state distribution ρ_0. For a policy π, reward discount $\gamma \in [0, 1)$, and episode length T, define

$$J_r(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{T-1} \gamma^t r_t \right], \qquad J_c(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{T-1} c_t \right]. \tag{1}$$

The objective is to maximize $J_r(\pi)$ subject to $J_c(\pi) \leq d$, where $d > 0$ is the episodic budget. A Lagrangian formulation uses $\mathcal{J}(\pi, \lambda) = J_r(\pi) - \lambda\big(J_c(\pi) - d\big), \lambda \geq 0$. RADPO instead uses an adaptive safety multiplier to scale predictive reward shaping, instead of directly optimizing $\mathcal{J}(\pi, \lambda)$.

Diffusion Policies and Policy Mirror Descent A diffusion policy $\pi_\theta(a \mid s)$ generates actions through iterative denoising (Ho et al., 2020). We denote its state-conditioned denoising loss by $\ell_{\text{diff}}(\theta; s, a)$. Let $Q(s, a)$ denote the action-value function used for policy improvement. For a reference policy π_{old}, KL-regularized $\pi_{\text{MD}}(a \mid s) = \frac{\pi_{\text{old}}(a|s) \exp(Q(s,a)/\tau)}{Z(s)}$, where $\tau > 0$ is the temperature and $Z(s) = \mathbb{E}_{a \sim \pi_{\text{old}}}[\exp(Q(s, a)/\tau)]$. This update motivates value-weighted denoising regression (Ma et al., 2025):

$$\mathcal{L}_{\text{MD}}(\theta) = \mathbb{E}_{\substack{s \sim \mu \\ a \sim \pi_{\text{old}}(\cdot|s)}} \left[\frac{\exp(Q(s, a)/\tau)}{Z(s)} \ell_{\text{diff}}(\theta; s, a) \right], \tag{2}$$

where μ is the state-sampling distribution. Actions and weights are detached during regression, avoiding differentiation through critics or the reverse diffusion sampler.

Discounted Cost Reachability Define the binary cost indicator $h_t = \mathbf{1}\{c_t > 0\}$. For a fixed policy π and discount $\gamma_c \in [0, 1)$, discounted first-hit reachability satisfies

$$V^\pi_{\text{reach}}(s_t) = \mathbb{E}_\pi[h_t + (1 - h_t)(1 - d_t)\gamma_c V^\pi_{\text{reach}}(s_{t+1}) \,|\, s_t], \tag{3}$$

where d_t indicates termination and terminal values are zero. This value weights the first future cost by γ_c raised to its arrival time, with zero contribution if no cost occurs before termination. It therefore captures both cost likelihood and timing, while remaining distinct from J_c, which counts cost magnitudes and repeated violations.

4 Reachability-Aware Diffusion Policy Optimization

RADPO combines predictive safety shaping with value-weighted diffusion policy optimization. A learned reachability critic predicts future cost exposure, while an adaptive safety multiplier controls the shaped penalty strength using realized episodic costs. The diffusion actor learns from candidates scored by the resulting reward critic. We describe these components below and describe some theoretical properties. All proofs for theoretical results are provided in Appendix A. A simplified illustrative diagram is available in Figure 1.B and the full pseudocode is available in Appendix F.2.

4.1 Predictive Safety Shaping

Standard cost critics propagate future violations backward through Bellman updates, thus safe states leading to unsafe outcomes can already receive high predicted cost. However, cumulative cost does not explicitly distinguish how soon the next cost may be incurred. RADPO instead shapes the reward using discounted first-hit reachability, which is higher when constraint violation is more likely or expected to occur sooner. Repeatedly applying this penalty along the trajectory can thus discourage behavior that approaches violations, even when the immediate cost is zero.

We learn an ensemble of bounded reachability critics $V_{reach,m}(s) \in [0,1]$, $m = 1, \dots, M$. For a replay transition $(s_t, a_t, r_t, c_t, s_{t+1}, d_t)$, each head minimizes binary cross entropy against the target

$$y^c_{t,m} = h_t + (1 - h_t)(1 - d_t)\gamma_c V^-_{reach,m}(s_{t+1}), \tag{4}$$

where $V^-_{reach,m}$ denotes its target network and $h_t = \mathbf{1}\{c_t > 0\}$. The following proposition describes the exact fixed-policy value counterpart of this update.

Proposition 1 (First-hit interpretation)**.** *Fix a stationary policy π and $0 < \gamma_c < 1$. Let*

$$\tau_H = \inf\{k \geq 0 : h_k = 1\}, \qquad \inf \varnothing := \infty,$$

and let $\tau_{\text{end}} \in \mathbb{N} \cup \{\infty\}$ denote the number of transitions before termination, with $\tau_{\text{end}} = \infty$ for a nonterminating trajectory. Then the unique bounded fixed point of Equation 3 satisfies $0 \leq V^\pi_{\text{reach}}(s) \leq 1$ and is

$$V^\pi_{\text{reach}}(s) = \mathbb{E}^\pi_s[\gamma_c^{\tau_H}\mathbf{1}\{\tau_H < \tau_{\text{end}}\}]\,. \tag{5}$$

Proposition 1 shows that discounted first-hit reachability combines the likelihood and timing of a future constraint violation. A smaller value can reflect either a lower violation probability or a more distant violation. The proof is given in Appendix A.1.

The shaping signal uses the online ensemble mean $V_{reach}(s) = \frac{1}{M}\sum_{m=1}^{M} V_{reach,m}(s)$:

$$\widetilde{r}_t = r_t - \beta\lambda V_{reach}(s_{t+1}), \tag{6}$$

where $\beta > 0$ is a fixed shaping coefficient, and $\lambda \geq 0$ is the adaptive safety multiplier (Sec. 4.3). Twin reward critics are trained against

$$y^r_t = \widetilde{r}_t + (1 - d_t)\gamma Q^-_r(s_{t+1}, a_{t+1}), \tag{7}$$

where $Q^-_r(s,a) = \min_{j\in\{1,2\}} Q^-_{r,j}(s,a)$ and a_{t+1} denotes the bootstrap action. The actor uses no separate cost term in its reweighting logits; safety is incorporated through the shaped reward.

Relation to cumulative cost. To understand how reachability shaping relates to the original safety objective, consider exact evaluation under a fixed policy π. Define accumulated reachability and discounted cumulative cost:

$$G^\pi_{\gamma,\gamma_c}(s,a) = \mathbb{E}^\pi_{s,a}\left[\sum_{k\geq 0}\gamma^k V^\pi_{\text{reach}}(S_{t+k+1})\right],\ Q^{\pi,\delta}_c(s,a) = \mathbb{E}^\pi_{s,a}\left[\sum_{k\geq 0}\delta^k c_{t+k}\right],\ 0 < \delta < 1, \tag{8}$$

with zero continuation after termination.

Theorem 1 (Conservative cumulative-cost surrogate). *Assume* $0 \le c_t \le c_{\max} h_t$, *with* $c_{\max} > 0$, *and* $\delta \le \gamma < 1$. *Then*

$$Q_c^{\pi,\delta}(s,a) \le \mathbb{E}[c_t \mid s,a] + c_{\max}\delta G^{\pi}_{\gamma,\gamma_c}(s,a). \tag{9}$$

If $c_t = 0$ *almost surely for the initial* (s,a), *then*

$$G^{\pi}_{\gamma,\gamma_c}(s,a) \ge \frac{Q_c^{\pi,\delta}(s,a)}{c_{\max}\delta}. \tag{10}$$

Theorem 1 shows that, after separating the immediate cost, the accumulated reachability signal provides a conservative surrogate for future discounted cumulative cost: a large future cost must be accompanied by a correspondingly large accumulated reachability penalty, up to a fixed factor. When $c_t = 0$, we obtain a bound relating $G^{\pi}_{\gamma,\gamma_c}$ with $Q_c^{\pi,\delta}$. Consequently, reachability can also assign predictive penalties to safe predecessor states before the cost is incurred. The proof is in Appendix A.2.

Each reachability value depends only on the first subsequent cost, but RADPO applies this signal repeatedly along the trajectory. To show how this relates to repeated cost exposure, let

$$D_h^{\pi,+}(s,a) = \mathbb{E}^{\pi}_{s,a}\left[\sum_{k\ge 1} \gamma^{k-1} h_{t+k}\right]. \tag{11}$$

Corollary 1 (Repeated cost exposure). *If* $0 \le \gamma_c < \gamma < 1$, *then*

$$D_h^{\pi,+}(s,a) \le G^{\pi}_{\gamma,\gamma_c}(s,a) \le \frac{D_h^{\pi,+}(s,a)}{1-\gamma_c/\gamma}. \tag{12}$$

Corollary 1 shows that accumulated reachability is bounded above and below by discounted repeated cost exposure, up to a fixed scaling factor. Thus, although each reachability value is first-hit, repeatedly applying it preserves sensitivity to multiple future violations. Its proof is in Appendix A.3.

4.2 Diffusion Policy Improvement by Candidate Reweighting

Critic regression uses replay state-action pairs, whereas actor targets are generated at replay next states s'. For each s', we sample K candidates $\bar{a}_i \sim \pi_\theta(\cdot \mid s')$ and perturb them before scoring, i.e. $a_i = \Pi_{\mathcal{A}}(\bar{a}_i + \epsilon_i), \epsilon_i \sim \mathcal{N}(0, \sigma_{\exp}^2 I)$, where $\Pi_{\mathcal{A}}$ clips actions to their bounds. The exploration scale is $\sigma_{\exp} = \sigma_0 \exp(\xi)$, with fixed σ_0 and a learned log-scale ξ adapted toward a target Gaussian entropy.

Candidates receive scores $q_i = \min_{j\in\{1,2\}} Q_{r,j}(s', a_i)$. Let $\bar{q} = \frac{1}{K}\sum_i q_i$ and $\sigma = \max(\sigma_A, 10^{-6})$, where σ_A is the advantage-normalization scale. We center, normalize, and clip the scores: $z_i = \mathrm{clip}\big(\frac{q_i - \bar{q}}{\sigma}, -3, 3\big)$, $\widehat{A}_i = z_i/\tau$, where $\tau > 0$ controls the reweighting strength. The regression weights are $w_i = K\frac{\exp(\widehat{A}_i)}{\sum_j \exp(\widehat{A}_j)}$, $\frac{1}{K}\sum_i w_i = 1$. The actor minimizes the weighted denoising objective

$$\mathcal{L}_\pi(\theta) = \mathbb{E}\left[\frac{1}{K}\sum_{i=1}^{K} w_i\, \ell_{\mathrm{diff}}(\theta; s', a_i)\right], \tag{13}$$

where the expectation covers replay successor states and candidate generation. Candidates and weights are detached, thus the update requires neither action gradients through critics nor differentiation through the reverse diffusion sampler.

Finite-candidate improvement. Building on the policy-mirror-descent interpretation underlying DPMD (Ma et al., 2025), we describe the finite-candidate reweighting used by RADPO. Although the subsequent diffusion regression is approximate, candidate reweighting step itself admits a mirror-descent interpretation over the sampled actions.

Theorem 2 (Finite-candidate mirror descent). *For a fixed candidate set, let* $p_i = w_i/K$, $u_i = 1/K$, *and let* Δ_K *denote the probability simplex. Then*

$$p = \arg\max_{v\in\Delta_K}\left\{\sum_i v_i z_i - \tau D_{\mathrm{KL}}(v\|u)\right\}. \tag{14}$$

Moreover,

$$\sum_i p_i q_i - \frac{1}{K}\sum_i q_i \geq \tau\sigma\left[D_{\mathrm{KL}}(p\|u) + D_{\mathrm{KL}}(u\|p)\right] \geq 0. \tag{15}$$

Theorem 2 shows that the weights implement a KL-regularized mirror-descent update over the sampled candidates, starting from uniform weights. Moreover, despite normalization and clipping of the scores used to compute the weights, reweighted candidates cannot have a lower mean reward-critic score than the original candidate set. Safety affects this improvement through the reward critic, which is trained on the reachability-shaped reward. The proof is given in Appendix A.4.

4.3 Adaptive Safety Multiplier

The reachability signal predicts first-hit cost exposure, while the safety-multiplier controller uses the original undiscounted episodic costs, including their magnitudes and repeated violations. Let $\widehat{J}_c$ denote a recent-window estimate of episodic cost constructed from completed training episodes, as detailed in Appendix F.1. The safety multiplier follows the projected update

$$\lambda \leftarrow \Pi_{[0,\lambda_{\max}]}\left[\lambda + \eta_\lambda \frac{\widehat{J}_c - d}{d}\right], \tag{16}$$

where $\eta_\lambda > 0$ is the controller step size. The multiplier increases when observed cost exceeds the budget and decreases below budget, subject to projection.

Budget-tracking property. The controller admits a pathwise bound on the average episodic-cost statistic supplied to it, without requiring convergence of the actor or critics.

Proposition 2 (Average observed-budget tracking)**.** *Let $\lambda_0 \in [0, \lambda_{\max}]$ and let $\widehat{J}_{c,j}$ denote the controller statistic at safety-multiplier update j. If upper clipping remains inactive for the first N updates, then*

$$\frac{1}{N}\sum_{j=0}^{N-1} \widehat{J}_{c,j} \leq d + \frac{d(\lambda_{\max} - \lambda_0)}{\eta_\lambda N}. \tag{17}$$

If upper clipping remains inactive for all updates, then

$$\limsup_{N\to\infty} \frac{1}{N}\sum_{j=0}^{N-1} \widehat{J}_{c,j} \leq d. \tag{18}$$

Proposition 2 shows that, as long as the upper bound is not active, sustained average budget violations cannot persist: the average controller statistic exceeds d by at most an $\mathcal{O}(1/N)$ term and is asymptotically no greater than d. Note that this result applies only to the statistic supplied to the controller and does not imply per-policy or per-episode feasibility. The proof and the corresponding bound when upper clipping occurs are given in Appendix A.5.

5 Experiments

5.1 Experimental Setup

We evaluate RADPO on ten continuous-control safety tasks in Safety Gymnasium (Ji et al., 2024) against CAL (Wu et al., 2024), SAC-Lag, PPO-Lag, RESPO (Ganai et al., 2023), ALGD (Cheng et al., 2026)[1] and an unconstrained reward-only RADPO ablation. The experimental protocol evaluates six random seeds per configuration. The episodic cost budget is set to $d = 25$, except for RESPO, which still enforces a zero-cost threshold by its formulation but is included as a hard-safety baseline. We report the mean and standard deviation of episodic return and episodic cost across seeds. The full list of hyperparameters is available in Appendix H.

[1]We compare against ALGD using the authors' released code, which includes Lagrangian energy itself as an auxiliary loss term not described in the paper. Removing it prevents useful learning in our reproduction, therefore we report the released implementation and note this discrepancy.

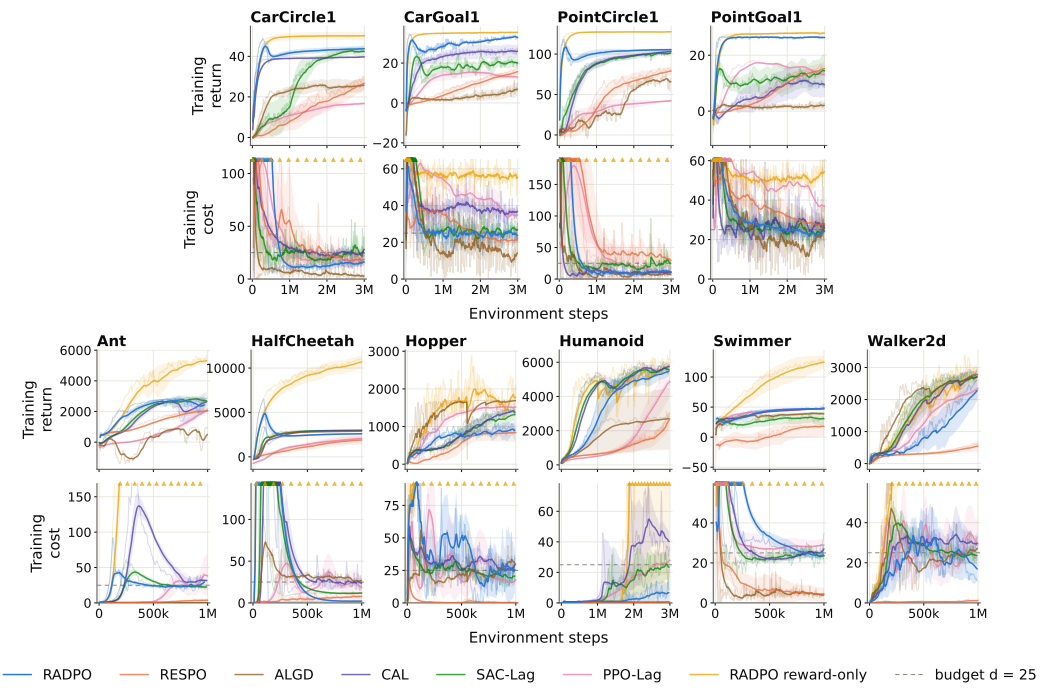

Figure 2: **Training performance across ten continuous-control safety tasks.** Episodic return (top rows) and episodic cost (bottom rows) for RADPO and baseline methods. Curves show empirical means across six random seeds; shaded regions denote one standard deviation. Dashed horizontal lines indicate the safety budget $d = 25$.

5.2 Main Results

Figure 2 shows that RADPO substantially reduces constraint violations relative to reward-only RADPO while maintaining competitive returns, especially on the navigation tasks. The reward-only variant attains high returns but generally incurs severe constraint violations. Figure 3 summarizes the corresponding late-training reward–cost trade-offs. RADPO achieves both higher mean return and lower mean cost than the evaluated baselines on the two Circle tasks, and the highest mean return on both Goal tasks. On the Velocity tasks, RADPO is non-dominated on HalfCheetah, Humanoid, Swimmer, and Walker2d, while baselines attain better operating points on Ant and Hopper. Overall, RADPO provides favorable trade-offs across most evaluated tasks.

5.3 Ablation Studies

We investigate two key design decisions in adapting diffusion policies to safe reinforcement learning: the choice of safety signal and the structural placement of the safety penalty. Using CarGoal1, PointCircle1, and HalfCheetah, we compare RADPO against three ablations: (1) replacing reachability estimation with a standard discounted cost critic, (2) applying cost-only reward shaping, and (3) enforcing a Lagrangian safety penalty directly within candidate action weights. Additional ablation studies for hyperparameter values are available in Appendix C.

Safety-signal formulation. On CarGoal1, RADPO recovers target task returns while maintaining asymptotic costs close to the constraint boundary. Cost-only shaping yields higher returns but consistently breaches the safety limit, whereas discounted cost estimates result in worse returns and elevated violation rates. On PointCircle1, all three variants reach comparable asymptotic returns, but RADPO maintains lower late-training violation rates. On HalfCheetah, RADPO enforces the most conservative behavior, achieving the lowest cost among all variants at the expense of peak return. These results indicate that predictive reachability shaping offers an effective middle ground on obstacle avoidance tasks and robust constraint satisfaction in high-dimensional locomotion.

Penalty placement in diffusion optimization. Penalizing candidate action weights directly via Lagrangian multipliers causes severe optimization instability: returns plummet sharply before par-

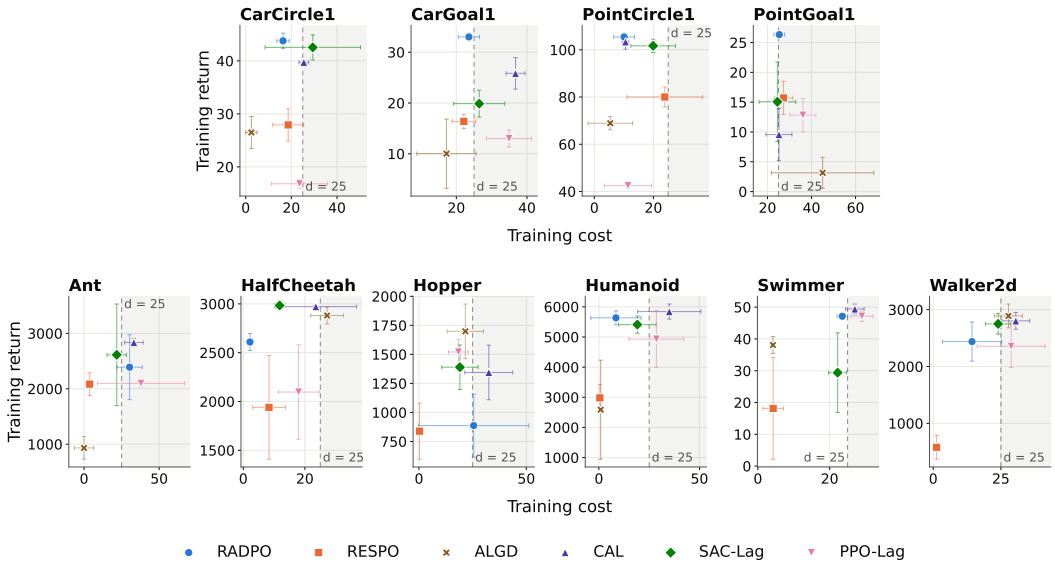

Figure 3: **Late-training return versus safety cost.** Markers indicate mean performance across seeds, computed over the final ten logged training points per run. Error bars represent one standard deviation across seed averages. Optimal performance lies toward the top-left quadrant (high return, low cost). Vertical dashed lines denote the budget $d = 25$ and the shaded region is infeasible.

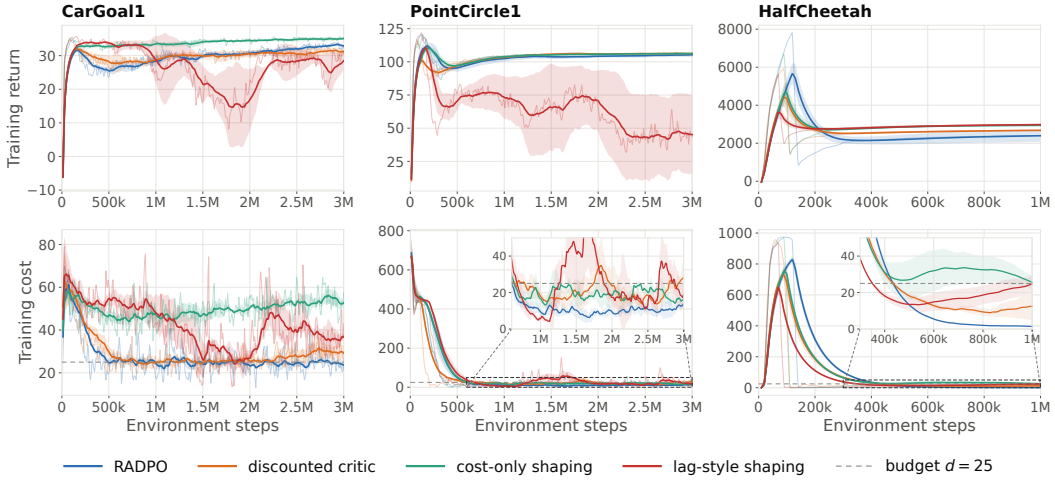

Figure 4: **Ablations on safety-signal formulation and penalty placement.** Training return (top) and episodic cost (bottom). Curves depict means across evaluated seeds with one standard deviation shaded. Solid lines show smoothing over raw trajectories (faint curves). Inset axes provide magnified views around the cost budget $d = 25$.

tially recovering on CarGoal1, and drop permanently on PointCircle1. In contrast, incorporating safety into reward shaping stabilizes policy learning, yielding superior returns and tighter constraint adherence across both domains. On HalfCheetah, candidate-weight penalization preserves higher returns while operating near the constraint boundary.

6 Discussion

Reachability interpretation. RADPO uses reachability as a policy-conditional prediction of cost exposure rather than as a certificate that costs can be avoided under an optimal controller. Proposition 1 shows that the value combines the probability of a positive-cost event with the timing of its first occurrence. Consequently, a smaller value can indicate either a less likely or a later violation, and should not be interpreted as an undiscounted violation probability. The discount γ_c controls this

temporal emphasis. RADPO therefore uses reachability as a predictive learning signal rather than as a safety certificate, since such guarantees would generally require additional assumptions in the model-free setting.

Separating reachability and budget feedback. A first-hit value does not encode violation severity or the number of subsequent violations, hence it cannot directly represent an episodic cost budget. RADPO instead uses the original episodic costs to adapt λ, while reachability determines the predictive penalty used to shape the reward. Repeated shaping nevertheless remains sensitive to repeated exposure: Equation 21 decomposes accumulated reachability into discounted cost exposure and an additional nonnegative predictive term assigned to safe predecessor states. Theorem 1 and Corollary 1 further relate this accumulated signal to cumulative cost and repeated violations under exact policy evaluation. Budget feedback adjusts the strength of this surrogate but does not recover the cost-magnitude information discarded by binarization. Because λ is updated from episodic cost but scales a reachability-based surrogate, it acts as a feedback multiplier rather than as a standard dual variable for the actor's objective.

Constraint structure and predictive reachability. We expect reachability shaping to be most useful when violations can be anticipated and avoided before cost is incurred, as in localized obstacle avoidance. In such settings, first-hit risk can distinguish trajectories that approach costs from those that maintain greater clearance, even when both have low immediate cost. The behavior in Figure 1 is consistent with this interpretation. Its benefit may diminish when unsafe behavior persists for many steps, thus cumulative cost already provides a strong signal, or when safety directly conflicts with reward. The stronger CarGoal1 benefit and the smaller HalfCheetah effect in Figure 4 are consistent with this hypothesis.

Limitations. Our theoretical analysis establishes properties of individual RADPO components rather than an end-to-end feasibility result. Such a result would require stronger assumptions, for example on function approximation, data coverage, and optimization convergence, than we make here for RADPO. In particular, the reachability analysis assumes exact fixed-policy evaluation, whereas the implemented state-only reachability critic is deliberately trained from replay for off-policy data reuse, so its population target reflects the replay behavior mixture rather than exactly the current policy. Likewise, the candidate-improvement guarantee applies before approximate diffusion regression, and the multiplier bound applies only to the observed cost statistic. Reachability remains a surrogate rather than an exact representation of the CMDP cost. Successor-state shaping does not directly penalize the current transition cost, and first-hit shaping changes the temporal weighting of violations. For nonbinary costs, $h_t = \mathbf{1}\{c_t > 0\}$ also discards magnitude information, although per-step costs are binary in all ten tasks considered here. Finally, budget regulation is indirect: episodic cost adjusts λ, which scales reachability shaping and influences the policy only through subsequent critic and actor updates. Transient violations and sensitivity to the controller timescale or multiplier cap can therefore still occur, as in primal-dual methods.

7 Conclusion

We introduced Reachability-Aware Diffusion Policy Optimization (RADPO), a model-free approach to the relatively unexplored setting of safe online diffusion-policy optimization under cumulative-cost constraints. RADPO combines predictive first-hit reachability with episodic budget feedback, while retaining candidate reweighting without dynamics models, critic action gradients, or reverse-process differentiation. Our theoretical analysis relates accumulated reachability to cumulative cost and repeated violations, and our experiments show substantial reductions in constraint violations on several tasks while maintaining competitive returns.

AI Use Statement

We used Generative AI tools to assist with parts of the conceptual and methodological development, mathematical claim formulation and refinement, and proof development and writing, as well as with experimental-design feedback and software implementation and debugging. In addition, they were used for literature search and summarization, manuscript drafting, and editing for typos, clarity and structure. AI tools were not used to propose or refine research hypotheses, interpret experimental

results, generate, clean, or reformat datasets, perform translation, or conduct qualitative or thematic data analysis. All AI-assisted content, including mathematical arguments, code, and citations, was reviewed and verified by the authors. We take responsibility for the final content of our work.

REPRODUCIBILITY STATEMENT

Complete proofs of the theoretical results are provided in Appendix A. Task definitions are given in Appendix B, algorithmic and implementation details in Appendix F, and the complete experimental protocols, hyperparameters, baseline adaptations, and ablation configurations in Appendix H. We additionally provide the code used for the experiments as supplementary material.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization, 2017. URL https://arxiv.org/abs/1705.10528.

Anayo K. Akametalu, Shromona Ghosh, Jaime F. Fisac, Vicenc Rubies-Royo, and Claire J. Tomlin. A minimum discounted reward hamilton-jacobi formulation for computing reachable sets. *IEEE Transactions on Automatic Control*, 69(2):1097–1103, February 2024. ISSN 0018-9286. doi: 10.1109/TAC.2023.3327159. Publisher Copyright: © 1963-2012 IEEE.

Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J. Tomlin. Hamilton-jacobi reachability: A brief overview and recent advances, 2017. URL https://arxiv.org/abs/1709.07523.

Rui Chen, Shangtao Li, Yifan Sun, and Changliu Liu. λ-reachability: Geometric-horizon safety bellman equations for humanoid safety, 2026. URL https://arxiv.org/abs/2606.16022.

Xiaoyuan Cheng, Wenxuan Yuan, Boyang Li, Yuanchao Xu, Yiming Yang, Hao Liang, Bei Peng, Robert Loftin, Zhuo Sun, and Yukun Hu. How does the lagrangian guide safe reinforcement learning through diffusion models? In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=mK3xWE7p2T.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion, 2024. URL https://arxiv.org/abs/2303.04137.

Kutay Demiray, Efe Eren Ceyani, and Ozgur S. Oguz. Uncertainty-aware safety propagation critics for safe reinforcement learning. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=b8GWqgLEHh.

Shutong Ding, Ke Hu, Zhenhao Zhang, Kan Ren, Weinan Zhang, Jingyi Yu, Jingya Wang, and Ye Shi. Diffusion-based reinforcement learning via q-weighted variational policy optimization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.16173.

Jaime F. Fisac, Anayo K. Akametalu, Melanie N. Zeilinger, Shahab Kaynama, Jeremy Gillula, and Claire J. Tomlin. A general safety framework for learning-based control in uncertain robotic systems, 2018. URL https://arxiv.org/abs/1705.01292.

Milan Ganai, Zheng Gong, Chenning Yu, Sylvia Herbert, and Sicun Gao. Iterative reachability estimation for safe reinforcement learning, 2023. URL https://arxiv.org/abs/2309.13528.

Chen-Xiao Gao, Chenyang Wu, Mingjun Cao, Chenjun Xiao, Yang Yu, and Zongzhang Zhang. Behavior-regularized diffusion policy optimization for offline reinforcement learning, 2025. URL https://arxiv.org/abs/2502.04778.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020.

Jiaming Ji, Borong Zhang, Jiayi Zhou, Xuehai Pan, Weidong Huang, Ruiyang Sun, Yiran Geng, Yifan Zhong, Juntao Dai, and Yaodong Yang. Safety-gymnasium: A unified safe reinforcement learning benchmark, 2024. URL https://arxiv.org/abs/2310.12567.

Huy Le, Onur Celik, Denis Blessing, Tai Hoang, Claas A Voelcker, Axel Brunnbauer, Felix Richter, Michael Volpp, and Gerhard Neumann. Trust-region diffusion policies for massively parallel on-policy RL. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=mGu2fs7kJt.

Jiaxing Li, Hanjiang Hu, Yujie Yang, and Changliu Liu. Verifiable safety q-filters via hamilton-jacobi reachability and multiplicative q-networks, 2025. URL https://arxiv.org/abs/2506.15693.

Qian Lin, Bo Tang, Zifan Wu, Chao Yu, Shangqin Mao, Qianlong Xie, Xingxing Wang, and Dong Wang. Safe offline reinforcement learning with real-time budget constraints, 2024. URL https://arxiv.org/abs/2306.00603.

Ben Liu, Shunpeng Yang, and Hua Chen. Diffusion policy through conditional proximal policy optimization, 2026. URL https://openreview.net/forum?id=sOSdvn2sM2.

Zuxin Liu, Zhepeng Cen, Vladislav Isenbaev, Wei Liu, Zhiwei Steven Wu, Bo Li, and Ding Zhao. Constrained variational policy optimization for safe reinforcement learning, 2022. URL https://arxiv.org/abs/2201.11927.

Haitong Ma, Tianyi Chen, Kai Wang, Na Li, and Bo Dai. Efficient online reinforcement learning for diffusion policy. *arXiv preprint arXiv:2502.00361*, 2025.

Nikola Milosevic, Johannes Müller, and Nico Scherf. Embedding safety into RL: A new take on trust region methods. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=4zRb89SbzG.

I.M. Mitchell, A.M. Bayen, and C.J. Tomlin. A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50(7): 947–957, 2005. doi: 10.1109/TAC.2005.851439.

Kazuki Mizuta and Karen Leung. Cobl-diffusion: Diffusion-based conditional robot planning in dynamic environments using control barrier and lyapunov functions. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024.

Michael Psenka, Alejandro Escontrela, Pieter Abbeel, and Yi Ma. Learning a diffusion model policy from rewards via q-score matching. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Allen Z. Ren, Justin Lidard, Lars L. Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. In *arXiv preprint arXiv:2409.00588*, 2024.

Hikmet Simsir and Ozgur S. Oguz. Lagrangian perturbation diffusion steering: Latent reinforcement learning for generative policies, 2026. URL https://arxiv.org/abs/2606.01151.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by pid lagrangian methods, 2020. URL https://arxiv.org/abs/2007.03964.

Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. Reward constrained policy optimization, 2018. URL https://arxiv.org/abs/1805.11074.

Yinuo Wang, Likun Wang, Yuxuan Jiang, Wenjun Zou, Tong Liu, Xujie Song, Wenxuan Wang, Liming Xiao, Jiang Wu, Jingliang Duan, et al. Diffusion actor-critic with entropy regulator. *Advances in Neural Information Processing Systems*, 37:54183–54204, 2024.

Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=AHvFDPi-FA.

Zifan Wu, Bo Tang, Qian Lin, Chao Yu, Shangqin Mao, Qianlong Xie, Xingxing Wang, and Dong Wang. Off-policy primal-dual safe reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=vy42bYs1Wo.

Wei Xiao, Tsun-Hsuan Wang, Chuang Gan, Ramin Hasani, Mathias Lechner, and Daniela Rus. Safediffuser: Safe planning with diffusion probabilistic models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ig2wk7kK9J.

Long Yang, Zhixiong Huang, Fenghao Lei, Yucun Zhong, Yiming Yang, Cong Fang, Shiting Wen, Binbin Zhou, and Zhouchen Lin. Policy representation via diffusion probability model for reinforcement learning, 2023. URL https://arxiv.org/abs/2305.13122.

Tsung-Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J. Ramadge. Projection-based constrained policy optimization, 2020. URL https://arxiv.org/abs/2010.03152.

Dongjie Yu, Haitong Ma, Shengbo Eben Li, and Jianyu Chen. Reachability constrained reinforcement learning, 2022. URL https://arxiv.org/abs/2205.07536.

Jichen Zhang, Liqun Zhao, Antonis Papachristodoulou, and Jack Umenberger. Constrained diffusers for safe planning and control. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=tahkGZjjWA.

Yiming Zhang, Quan Vuong, and Keith W. Ross. First order constrained optimization in policy space, 2020. URL https://arxiv.org/abs/2002.06506.

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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=j5JvZCaDM0.

Guowei Zou, Weibing Li, Hejun Wu, Yukun Qian, Yuhang Wang, and Haitao Wang. D^{2}PPO: Diffusion policy policy optimization with dispersive loss. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, pp. 18891–18899, 2026. doi: 10.48448/e05v-4059. URL https://ojs.aaai.org/index.php/AAAI/article/download/38959/42921.

A Theoretical Analysis

A.1 Discounted reachability as a stopping-time value

For a fixed policy π, define the Bellman operator

$$(\mathcal{T}^\pi U)(s) := \mathbb{E}_\pi \left[h_0 + (1-h_0)(1-d_0)\gamma_c U(S_1) \,|\, S_0 = s \right]. \tag{19}$$

Lemma 1 (Contraction)**.** *For* $0 \le \gamma_c < 1$*,*

$$\|\mathcal{T}^\pi U - \mathcal{T}^\pi W\|_\infty \le \gamma_c \|U - W\|_\infty.$$

Proof. For every state s,

$$\begin{aligned}
&|(\mathcal{T}^\pi U)(s) - (\mathcal{T}^\pi W)(s)| \\
&= \gamma_c \left| \mathbb{E}_\pi \left[(1-h_0)(1-d_0)\big(U(S_1) - W(S_1)\big) \mid S_0 = s \right] \right| \\
&\le \gamma_c \mathbb{E}_\pi \left[|U(S_1) - W(S_1)| \mid S_0 = s \right] \\
&\le \gamma_c \|U - W\|_\infty.
\end{aligned}$$

Taking the supremum over s proves the claim. Since $\gamma_c < 1$, the operator is a contraction and therefore admits a unique bounded fixed point. □

Proof of Proposition 1. By Lemma 1, $\mathcal{T}^\pi$ has a unique bounded fixed point. Moreover, $\mathcal{T}^\pi$ maps functions with values in $[0,1]$ to functions with values in $[0,1]$. Iterating from the zero function therefore shows that

$$0 \le V^\pi_{\text{reach}}(s) \le 1.$$

Unrolling Equation 3 for n transitions gives

$$\begin{aligned}
V^\pi_{\text{reach}}(s) = \mathbb{E}^\pi_s \Bigg[&\sum_{k=0}^{n-1} \gamma_c^k h_k \prod_{j=0}^{k-1} (1-h_j)(1-d_j) \\
&+ \gamma_c^n V^\pi_{\text{reach}}(S_n) \prod_{j=0}^{n-1} (1-h_j)(1-d_j) \Bigg].
\end{aligned}$$

The absolute value of the final term is bounded by

$$\gamma_c^n \|V^\pi_{\text{reach}}\|_\infty,$$

which converges to zero as $n \to \infty$ because $0 < \gamma_c < 1$. Hence

$$V^{\pi}_{\text{reach}}(s) = \mathbb{E}^{\pi}_{s}\left[\sum_{k\geq 0} \gamma_c^k h_k \prod_{j=0}^{k-1}(1-h_j)(1-d_j)\right].$$

On any realized trajectory, if the first cost occurs at transition $\tau_H = k < \tau_{\text{end}}$, then all terms vanish except the one at k, which equals γ_c^k. If no cost occurs before termination, or if neither a cost nor termination ever occurs, every first-hit term is zero. Therefore

$$V^{\pi}_{\text{reach}}(s) = \mathbb{E}^{\pi}_{s}\left[\gamma_c^{\tau_H}\mathbf{1}\{\tau_H < \tau_{\text{end}}\}\right].$$

For the random-horizon interpretation, let L_{γ_c} be independent of the trajectory with

$$\Pr(L_{\gamma_c} = \ell) = (1-\gamma_c)\gamma_c^{\ell}, \qquad \ell = 0, 1, 2, \ldots.$$

Then

$$\Pr(L_{\gamma_c} \geq k) = \gamma_c^k.$$

Conditioning on $\tau_H = k$ therefore gives

$$\begin{aligned}
&\overset{\pi}{\Pr_s}\left(\tau_H \leq L_{\gamma_c},\ \tau_H < \tau_{\text{end}}\right) \\
&= \sum_{k\geq 0} \overset{\pi}{\Pr_s}\left(\tau_H = k,\ \tau_H < \tau_{\text{end}}\right)\Pr(L_{\gamma_c} \geq k) \\
&= \sum_{k\geq 0} \gamma_c^k \overset{\pi}{\Pr_s}\left(\tau_H = k,\ \tau_H < \tau_{\text{end}}\right) \\
&= V^{\pi}_{\text{reach}}(s).
\end{aligned}$$

□

Lemma 2 (Population consistency under fixed-policy sampling)**.** *Fix a stationary policy π and a target function $\bar{V} : \mathcal{S} \to [0,1]$. Suppose that, conditional on $S_t = s$, the transition (A_t, h_t, d_t, S_{t+1}) is generated under policy π and the environment dynamics. Define*

$$Y := h_t + (1-h_t)(1-d_t)\gamma_c \bar{V}(S_{t+1}).$$

For a fixed state s, the minimizer of the conditional population binary-cross-entropy loss

$$\mathbb{E}_{\pi}\left[-Y \log v - (1-Y)\log(1-v) \mid S_t = s\right]$$

is

$$v^*(s) = \mathbb{E}_{\pi}[Y \mid S_t = s] = (\mathcal{T}^{\pi}\bar{V})(s).$$

Proof. Let

$$m(s) := \mathbb{E}_{\pi}[Y \mid S_t = s].$$

Linearity of expectation gives

$$\mathcal{L}_s(v) = -m \log v - (1-m)\log(1-v).$$

Subtracting the value at $v = m$,

$$\mathcal{L}_s(v) - \mathcal{L}_s(m) = D_{\text{KL}}\big(\text{Ber}(m) \| \text{Ber}(v)\big) \geq 0.$$

Hence the population minimizer is $v = m$. Under the fixed-policy sampling assumption of the lemma, $m(s) = (\mathcal{T}^{\pi}\bar{V})(s)$ by Equation 19. □

Off-policy replay. Lemma 2 explains population regression under fixed-policy sampling. RADPO instead trains the state-only reachability critic from replay, consistent with its off-policy backbone. Let $\beta_{\mathcal{D}}(a \mid s)$ denote the effective conditional behavior distribution represented in the replay buffer. For fixed $\bar{V}$, the same BCE argument gives

$$v^*_{\mathcal{D}}(s) = \mathbb{E}_{\mathcal{D}}[Y \mid S_t = s] = (\mathcal{T}^{\beta_{\mathcal{D}}}\bar{V})(s),$$

which generally differs from $(\mathcal{T}^{\pi}\bar{V})(s)$ for the current policy. This is a deliberate trade-off: replay permits reuse of safety transitions and retains previously observed cost encounters that may become sparse as the policy improves. Since RADPO uses V_{reach} as a predictive shaping signal rather than as a safety certificate, the implementation favors this data reuse over exact on-policy evaluation. The resulting discrepancy should therefore be viewed as replay-distribution approximation error.

A.2 Conservative relation to cumulative cost

Proof of Theorem 1. Start from the cumulative cost value:

$$Q_c^{\pi,\delta}(s,a) = \mathbb{E}[c_t \mid s,a] + \sum_{k\geq 1} \delta^k \mathbb{E}[c_{t+k} \mid s,a].$$

Because

$$0 \leq c_{t+k} \leq c_{\max} h_{t+k},$$

we obtain

$$Q_c^{\pi,\delta}(s,a) \leq \mathbb{E}[c_t \mid s,a] + c_{\max} \sum_{k\geq 1} \delta^k \mathbb{E}[h_{t+k} \mid s,a]. \tag{20}$$

The reachability Bellman equation implies

$$V_{\text{reach}}^{\pi}(s) \geq \mathbb{E}[h_t \mid S_t = s],$$

because its continuation term is nonnegative. Applying the tower property,

$$\mathbb{E}\left[V_{\text{reach}}^{\pi}(S_{t+k}) \mid s,a\right] \geq \mathbb{E}[h_{t+k} \mid s,a].$$

Furthermore, $\delta \leq \gamma$ implies, for $k \geq 1$,

$$\delta^k = \delta\delta^{k-1} \leq \delta\gamma^{k-1}.$$

Hence

$$\begin{aligned}\sum_{k\geq 1} \delta^k \mathbb{E}[h_{t+k} \mid s,a] &\leq \delta \sum_{k\geq 1} \gamma^{k-1} \mathbb{E}\left[V_{\text{reach}}^{\pi}(S_{t+k}) \mid s,a\right] \\ &= \delta G_{\gamma,\gamma_c}^{\pi}(s,a).\end{aligned}$$

Substituting into Equation 20 proves Equation 9.

If $c_t = 0$ almost surely under (s,a), the immediate term vanishes and Equation 10 follows by rearranging. □

The source of the additional conservatism can be made explicit. Applying the reachability Bellman equation yields

$$G_{\gamma,\gamma_c}^{\pi}(s,a) = D_h^{\pi,+}(s,a) + \gamma_c P_{\text{pred}}^{\pi}(s,a), \tag{21}$$

$$P_{\text{pred}}^{\pi}(s,a) := \mathbb{E}_{s,a}^{\pi}\left[\sum_{k\geq 0} \gamma^k (1-h_{t+k+1})(1-d_{t+k+1}) V_{\text{reach}}^{\pi}(S_{t+k+2})\right] \geq 0.$$

Thus the difference between accumulated reachability and immediate discounted cost exposure consists of a nonnegative predictive term assigned to safe predecessor states.

A.3 Repeated cost exposure

Proof of Corollary 1. From the reachability Bellman recursion,

$$V_{\text{reach}}^{\pi}(S_{t+k}) \geq \mathbb{E}\left[h_{t+k} \mid S_{t+k}\right].$$

Taking conditional expectation from the initial (s,a), multiplying by the corresponding powers of γ, and summing gives

$$G_{\gamma,\gamma_c}^{\pi}(s,a) \geq D_h^{\pi,+}(s,a).$$

For the upper bound, first-hit reachability is pathwise bounded by the discounted sum of all future costs:

$$V_{\text{reach}}^{\pi}(S_n) \leq \mathbb{E}\left[\sum_{j\geq 0} \gamma_c^j h_{n+j} \,\middle|\, S_n\right].$$

Therefore

$$G^{\pi}_{\gamma,\gamma_c}(s,a) \le \sum_{k\ge 0}\sum_{j\ge 0} \gamma^k \gamma_c^j \mathbb{E}^{\pi}_{s,a}[h_{t+k+1+j}].$$

Set $u = k + 1 + j$. For each fixed $u \ge 1$, the coefficient multiplying $\mathbb{E}[h_{t+u}]$ is

$$\sum_{j=0}^{u-1} \gamma^{u-1-j}\gamma_c^j = \gamma^{u-1}\sum_{j=0}^{u-1}\left(\frac{\gamma_c}{\gamma}\right)^j.$$

Since $\gamma_c < \gamma$,

$$\sum_{j=0}^{u-1}\left(\frac{\gamma_c}{\gamma}\right)^j \le \frac{1}{1-\gamma_c/\gamma}.$$

Hence

$$G^{\pi}_{\gamma,\gamma_c}(s,a) \le \frac{1}{1-\gamma_c/\gamma} D_h^{\pi,+}(s,a),$$

which proves the result. □

A.4 Finite-candidate mirror descent

Lemma 3 (Empirical Gibbs variational identity)**.** *Let*

$$p_i = \frac{\exp(z_i/\tau)}{\sum_j \exp(z_j/\tau)}, \qquad u_i = \frac{1}{K}.$$

Then

$$p = \arg\max_{v\in\Delta_K} \left\{\langle v, z\rangle - \tau D_{\mathrm{KL}}(v\|u)\right\},$$

and

$$\langle p - u, z\rangle = \tau\left[D_{\mathrm{KL}}(p\|u) + D_{\mathrm{KL}}(u\|p)\right].$$

Proof. Let

$$Z_u := \frac{1}{K}\sum_i \exp(z_i/\tau).$$

Then

$$p_i = \frac{u_i \exp(z_i/\tau)}{Z_u},$$

so

$$\log\frac{p_i}{u_i} = \frac{z_i}{\tau} - \log Z_u.$$

For arbitrary $v \in \Delta_K$,

$$D_{\mathrm{KL}}(v\|p) = D_{\mathrm{KL}}(v\|u) - \frac{1}{\tau}\langle v, z\rangle + \log Z_u.$$

Rearranging,

$$\langle v, z\rangle - \tau D_{\mathrm{KL}}(v\|u) = \tau\log Z_u - \tau D_{\mathrm{KL}}(v\|p),$$

which is uniquely maximized at $v = p$.

Moreover,

$$\begin{aligned} &D_{\mathrm{KL}}(p\|u) + D_{\mathrm{KL}}(u\|p) \\ &= \sum_i (p_i - u_i)\log\frac{p_i}{u_i} \\ &= \frac{1}{\tau}\sum_i (p_i - u_i) z_i, \end{aligned}$$

because $\sum_i (p_i - u_i) = 0$. □

Proof of Theorem 2. The first statement follows directly from Lemma 3.

For the raw-Q bound, define

$$\sigma := \max(\sigma_A, 10^{-6}), \qquad R_i := q_i - \sigma z_i.$$

Then

$$q_i = \sigma z_i + R_i.$$

For fixed $\bar{q}$, the scalar function

$$R(q) = q - \sigma \operatorname{clip}\left(\frac{q - \bar{q}}{\sigma}, -3, 3\right)$$

is non-decreasing in q. The softmax probability p_i is also non-decreasing with q_i, because both clipping and exponentiation preserve order. Consequently,

$$(p_i - p_j)(R_i - R_j) \geq 0 \qquad \forall i, j.$$

Using the pairwise covariance identity,

$$\sum_i (p_i - u_i) R_i = \frac{1}{K} \sum_{i<j} (p_i - p_j)(R_i - R_j) \geq 0.$$

Therefore

$$\begin{aligned} \sum_i (p_i - u_i) q_i &= \sigma \sum_i (p_i - u_i) z_i + \sum_i (p_i - u_i) R_i \\ &\geq \sigma \sum_i (p_i - u_i) z_i. \end{aligned}$$

Applying Lemma 3,

$$\sum_i (p_i - u_i) q_i \geq \tau\sigma \left[D_{\mathrm{KL}}(p \| u) + D_{\mathrm{KL}}(u \| p) \right] \geq 0.$$

This is Equation 15. □

A.5 SAFETY-MULTIPLIER BUDGET TRACKING

For update index j, define the normalized observed budget error

$$e_j := \frac{\widehat{J}_{c,j} - d}{d}.$$

Define the amount discarded by upper clipping as

$$u_j := \left[\lambda_j + \eta_\lambda e_j - \lambda_{\max} \right]_+ .$$

Proposition 3 (Capped budget-tracking bound)**.** *For every $N \geq 1$,*

$$\frac{1}{N} \sum_{j=0}^{N-1} \widehat{J}_{c,j} \leq d + \frac{d}{\eta_\lambda N} \left[\lambda_{\max} - \lambda_0 + \sum_{j=0}^{N-1} u_j \right]. \tag{22}$$

Proof. Let

$$x_j := \lambda_j + \eta_\lambda e_j$$

be the unprojected update and define the lower-projection correction

$$\ell_j := [-x_j]_+.$$

Projection onto $[0, \lambda_{\max}]$ can be written exactly as

$$\lambda_{j+1} = \lambda_j + \eta_\lambda e_j + \ell_j - u_j.$$

Rearranging and summing,

$$\eta_\lambda \sum_{j=0}^{N-1} e_j = \lambda_N - \lambda_0 - \sum_{j=0}^{N-1} \ell_j + \sum_{j=0}^{N-1} u_j$$
$$\leq \lambda_{\max} - \lambda_0 + \sum_{j=0}^{N-1} u_j,$$

because $\lambda_N \leq \lambda_{\max}$ and $\ell_j \geq 0$. Substituting

$$e_j = \frac{\widehat{J}_{c,j} - d}{d}$$

and rearranging proves Equation 22. If the upper projection never activates, then $u_j = 0$ for all j, reducing Equation 22 exactly to Equation 17. □

At any converged interior fixed point, $0 < \lambda^* < \lambda_{\max}$, with limiting controller statistic $\widehat{J}_c^*$, the projection is inactive. Therefore

$$\lambda^* = \lambda^* + \eta_\lambda \frac{\widehat{J}_c^* - d}{d}$$

implies $\widehat{J}_c^* = d$. Boundary fixed points instead satisfy the corresponding one-sided conditions:

$$\lambda^* = 0 \Rightarrow \widehat{J}_c^* \leq d, \qquad \lambda^* = \lambda_{\max} \Rightarrow \widehat{J}_c^* \geq d.$$

These conditions resemble complementary slackness, but λ is not a standard Lagrange multiplier for the same objective optimized by the actor, consequently we use them only as an interpretation of the controller equilibrium.

B Task Details

We evaluate on ten environments from Safety Gymnasium (Ji et al., 2024): four safe-navigation tasks obtained by combining the Point and Car agents with the Goal1 and Circle1 tasks, and six velocity-constrained MuJoCo locomotion tasks.

Navigation agents. The **Point** agent is a simple planar robot with two control inputs: forward/backward motion and rotation about the vertical axis. The **Car** agent uses two independently actuated parallel wheels together with a free-rolling rear wheel, such that steering and forward motion must be coordinated through the two wheel controls. We evaluate both agents on the Goal1 and Circle1 tasks.

Navigation tasks. In **Goal1**, the agent is rewarded for reducing its distance to a target and receives an additional reward when the target is reached, after which a new target is generated. The Level-1 environment contains eight hazard regions and one movable vase; entering a hazard produces safety cost, whereas the vase is not part of the safety constraint. In **Circle1**, the agent is rewarded for moving around a prescribed circular path while remaining close to its radius. Two virtual boundary walls restrict the admissible region, and crossing from the interior to the exterior produces safety cost.

Velocity-constrained locomotion. The remaining six environments augment the standard Gymnasium MuJoCo locomotion tasks with a velocity constraint. We consider **Ant**, a 3D quadruped; **HalfCheetah**, a planar running robot; **Hopper**, a planar monoped; **Humanoid**, a 3D biped; **Swimmer**, an articulated swimming agent; and **Walker2d**, a planar biped. The underlying task rewards locomotion according to the corresponding MuJoCo environment, while Safety Gymnasium produces a binary cost whenever the agent's velocity exceeds an environment-specific threshold. For Swimmer the constraint is applied to the x-axis velocity; for the other agents the relevant planar velocity magnitude is used.

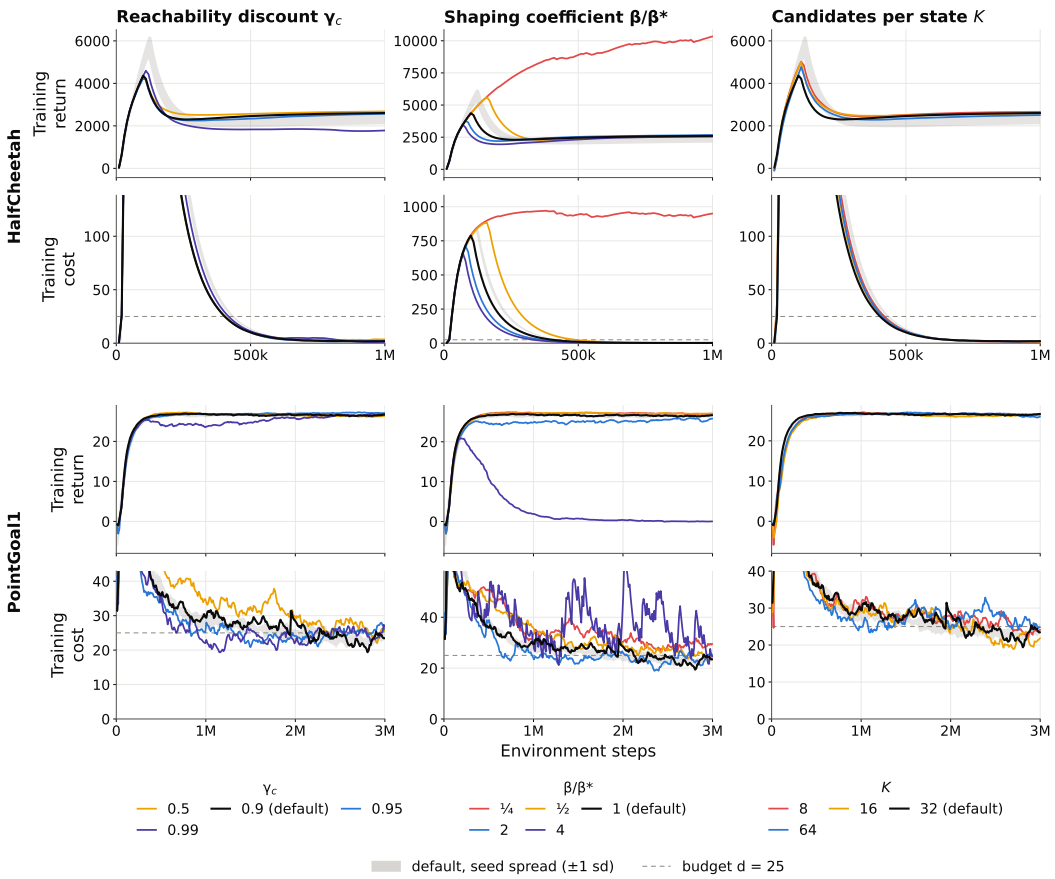

Figure 5: **Ablations on hyperparameter values.** Ablation results for reachability discount γ_c, shaping coefficient β and candidates per state K. Curves depict means across evaluated seeds with one standard deviation shaded.

C Additional Ablations

To justify our hyperparameter choices, we show our ablations for reachability discount γ_c, shaping coefficient β and candidates per state K in Figure 5. As expected, higher values of γ_c yield a more conservative policy, especially for HalfCheetah. Similarly, higher β results in a more conservative policy; for PointGoal1, when β is too high the reward signal is drowned out and the policy fails to learn. Finally for K, we only possibly see a slight improvement in safety for higher values in exchange for a higher computational cost.

D Computational Costs

Table 1 shows computational cost comparison of ALGD, Lagrangian-style shaping and RADPO on PointGoal1. Runs were taken on a computer with one RTX 3090 Ti and i9-12900F. Lagrangian-style and RADPO runs were taken with our codebase. ALGD uses the official implementation provided by their authors, modified only to use Safety Gymnasium and log the timing details shown in this table. Note that these runs retain the 10/0.25/0.25 actual learner updates per transition for ALGD, Lagrangian-style and RADPO respectively, as described in their default configurations, which means that ALGD has $40\times$ more updates per environment step, hence the significantly longer times.

E Additional Figures

Figure 6 shows the state distribution of the paths shown in Figure 1.

Method	Time per environment transition (ms)				
	Sample acquisition	Critic optimization	Diffusion-policy optimization	Other learning	Overall
ALGD	3.298	16.057	78.892	41.524	139.771
Lagrangian-style	0.418	0.545	0.459	3.387	4.809
RADPO	0.616	0.423	0.406	2.697	4.142

Table 1: Computational cost on PointGoal1. All entries are ms per environment step.

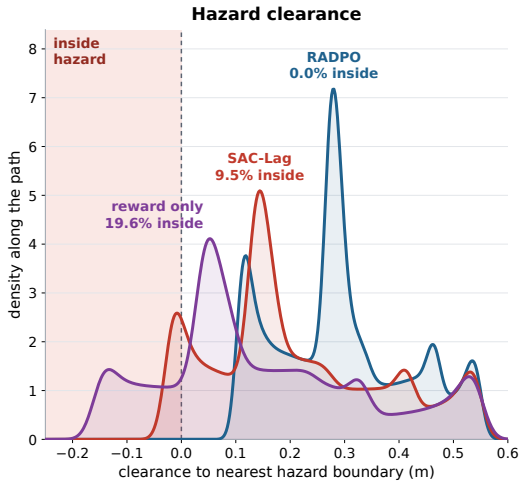

Figure 6: Distribution of states for trajectories in Figure 1.A.

F Implementation Details

F.1 Safety-multiplier Cost Estimate

The safety-multiplier controller operates on completed training episodes. For each collection event j with a nonempty set of completed episodes $\mathcal{E}_j$, we compute the mean undiscounted episodic cost

$$\bar{C}_j = \frac{1}{|\mathcal{E}_j|} \sum_{e \in \mathcal{E}_j} \sum_{t=0}^{L_e - 1} c_t^{(e)}. \tag{23}$$

The controller statistic averages these completion-batch means over a recent window $\mathcal{W}$:

$$\widehat{J}_c = \frac{1}{|\mathcal{W}|} \sum_{j \in \mathcal{W}} \bar{C}_j. \tag{24}$$

F.2 Pseudocode

We present the pseudocode for RADPO training in Algorithm 1.

G Extension to Graded Costs and Multiple Constraints

Since the RADPO algorithm as described assumes a single constraint with a binary per-step cost signal, in this section we discuss extension of RADPO to graded (continuous) costs and multiple constraint settings. Note that our Safety Gymnasium tasks have single constraints with binary costs by default.

G.1 Graded Costs

The bounded-cost assumption in Theorem 1 does not require binary costs. For any nonnegative graded cost $c_t \in [0, c_{\max}]$ and $h_t = \mathbf{1}\{c_t > 0\}$, the condition $c_t \leq c_{\max} h_t$ holds automati-

Algorithm 1 Reachability-Aware Diffusion Policy Optimization (RADPO)

Require: Cost budget d, candidate count K

Initialize actor, critics, target networks, replay buffer $\mathcal{D}$, and safety multiplier λ
while training **do**
  Collect experience using candidate selection and exploration
  Store transitions in $\mathcal{D}$
  Adapt λ using recent episodic costs — Equation 16
  for each gradient update **do**
    Sample (s, a, r, c, s', d_t) from $\mathcal{D}$
    Compute reachability targets — Equation 4
    Shape rewards: $\widetilde{r} = r - \beta\lambda V_{reach}(s')$
    Compute reward-critic targets — Equation 7
    Sample and perturb K actor candidates at s'
    Score candidates with Q_r and compute weights w_i
    Update reachability and reward critics
    Update actor using $\mathcal{L}_\pi = \mathbb{E}_{s'}[\frac{1}{K}\sum_{i=1}^{K} w_i\, \ell_{\text{diff}}(\theta; s', a_i)]$ with detached candidates and weights
    Update exploration scale, statistics, and targets as scheduled
  end for
end while

cally. Thus, the cumulative-cost bound continues to apply without modification. Binarization only removes cost-magnitude information from the predictive reachability signal; the original graded episodic costs are still retained by the safety-multiplier update. Hence RADPO can be applied directly to bounded graded costs, although its reachability shaping distinguishes the timing and likelihood of positive-cost events rather than their severity.

G.2 Multiple Constraints

For constraints $J_{c_i}(\pi) \le d_i$, $i = 1, \dots, m$, RADPO can be extended by maintaining a reachability critic and adaptive safety multiplier for each constraint. Each multiplier is updated from its corresponding realized episodic cost and budget, while reward shaping becomes

$$\widetilde{r}_t = r_t - \sum_{i=1}^{m} \beta_i \lambda_i V^i_{reach}(s_{t+1}). \tag{25}$$

The reachability and cumulative-cost relations of Theorem 1 then apply separately to each constraint. This preserves the individual budgets and predictive safety signals rather than collapsing multiple constraints into a single scalarized cost.

H Experimental Details and Hyperparameters

This appendix specifies every setting used to produce Figures 2–4. Section H.1 fixes the tasks, budgets and evaluation protocol shared by all methods; Section H.2 gives RADPO's hyperparameters, split into settings that are constant across all ten tasks and the small set that varies per task; Section H.3 gives the baseline configurations; and Section H.4 gives the ablation arms of Section 5.3.

H.1 Interaction budgets and evaluation protocol

All experiments use Safety Gymnasium (Ji et al., 2024): six MuJoCo velocity-constrained locomotion tasks and four navigation tasks, described in Appendix B. Every task uses an episode horizon of $T = 1000$ steps and the episodic cost budget $d = 25$, except where a baseline's own formulation prescribes otherwise (Section H.3). Per-step costs are binary on all ten tasks, so the binarisation $h_t = \mathbf{1}\{c_t > 0\}$ discards no magnitude information here.

Evaluation. Training curves report the episodic return and cost of the on-policy training rollouts; the latter are used to construct the safety-multiplier statistic. Deterministic evaluation runs 20

episodes at the cadence in Table 2. The late-training scatter of Figure 3 averages each run's final ten logged training points, then reports the mean and standard deviation over the runs of an arm.

Table 2: Tasks, interaction budgets and evaluation cadence.

Family	Task	Horizon T	Env. steps	Eval. every
Velocity	SafetyAntVelocity-v1	1000	1M	25k
	SafetyHalfCheetahVelocity-v1	1000	1M	25k
	SafetyHopperVelocity-v1	1000	1M	25k
	SafetyHumanoidVelocity-v1	1000	3M	25k
	SafetySwimmerVelocity-v1	1000	1M	25k
	SafetyWalker2dVelocity-v1	1000	1M	25k
Navigation	SafetyCarCircle1-v0	1000	3M	100k
	SafetyCarGoal1-v0	1000	3M	100k
	SafetyPointCircle1-v0	1000	3M	100k
	SafetyPointGoal1-v0	1000	3M	100k

Implementation and compute. All methods except RESPO and ALGD are implemented in a single JAX code base that shares the environment wrappers, replay buffer, vectorised sampler, evaluation loop and logging, hence cross-method differences come from the algorithms and not from the harness. RESPO and ALGD are run from their authors' released implementations. Runs were executed on a heterogeneous pool of consumer and workstation NVIDIA GPUs (GTX 1080, RTX 4090, RTX 5090, RTX 6000 Ada/Pro), one run per GPU.

H.2 RADPO

H.2.1 Settings shared by all ten tasks

Table 3 lists every RADPO hyperparameter that is identical on all ten tasks; Algorithm 1 in Appendix F.2 gives the update order these settings parameterise. The diffusion actor, the twin reward critics and the reachability critics are all 3×256 MLPs with Mish activations; the actor additionally takes the denoising-step embedding as input (Section 3). Updates are Adam. The actor's learning rate is linearly annealed from 3×10^{-4} to 3×10^{-5} between gradient steps 2.5×10^4 and 7.5×10^4; the critic learning rates are constant. The exploration scale is $\sigma_{\text{exp}} = \sigma_0 \exp(\xi)$, with ξ adapted every 250 gradient steps toward a fixed Gaussian entropy target proportional to the action dimension. Safety enters RADPO only through reachability shaping of the reward critic, $\widetilde{r}_t = r_t - \beta\lambda V_{reach}(s_{t+1})$, on every task; the candidate weights of Equation 4.2 carry no separate cost term.

H.2.2 Per-task settings

Table 4 lists the settings that vary across tasks. We tune the update-to-data ratio, initial exploration temperature α_0, exploration scale σ_0, shaping coefficient β, and controller step size η_λ.

Among these, η_λ is the least transferable across tasks. The update

$$\lambda \leftarrow \Pi_{[0,\lambda_{\max}]} \left[\lambda + \eta_\lambda \frac{\hat{J}_c - d}{d} \right]$$

scales with the relative budget violation $(\hat{J}_c - d)/d$, which depends on the task. For example, CarGoal1 requires $\eta_\lambda = 0.055$, whereas the Circle tasks use 0.005. Swimmer also requires a relatively large value, $\eta_\lambda = 0.10$, because its episodes do not terminate early. The controller, which updates only when episodes are completed, therefore receives substantially fewer updates during training, and a smaller step size adapts too slowly within the available interaction budget.

Table 3: RADPO hyperparameters held constant across all ten tasks.

Group	Hyperparameter	Value
Networks	Critic / reachability-critic MLP	3×256, Mish
	Diffusion actor MLP	3×256, Mish
	Denoising steps	20
	Noise schedule	cosine
	Reward critics (twin)	2
Optimisation	Optimiser	Adam
	Critic learning rate	3×10^{-4}
	Actor learning rate	$3 \times 10^{-4} \to 3 \times 10^{-5}$ (linear)
	Replay capacity	10^6 transitions
	Batch size	256
	Warm-up (uniform-random) transitions	30k
	Actor / target update period	every 2 gradient steps
	Target smoothing τ_{Polyak}	0.005
Candidate reweighting	Candidates per state K	32
	Reweighting temperature τ	1.0
	Score clip	$[-3, 3]$
	Advantage scale σ_A	EMA of batch std, coefficient 10^{-3}
Exploration	Entropy-target update period	every 250 gradient steps
	ξ learning rate	3×10^{-4}
	Action projection $\Pi_{\mathcal{A}}$	clip to box bounds
Safety	Shaping signal	$V_{reach}(s_{t+1})$ (reachability)
	Reachability ensemble size M	8
	Reachability loss	binary cross-entropy
	Reachability discount γ_c	0.9
	Reward discount γ	0.99
Multiplier controller	Budget d	25
	λ initialisation	0
	λ lower bound	0
	λ cap $\lambda_{\max}$	10
	Cost window $\lvert\mathcal{W}\rvert$	10 completion batches (App. F.1)
	Update trigger	each collection event with ≥ 1 completed episode
Evaluation	Deterministic episodes	20
	Candidate selection at deployment	highest reward-critic score

Table 4: Per-task RADPO hyperparameters. UTD is the number of optimiser updates per collected transition, α_0 the initial exploration temperature ($\sigma_{\text{exp}} = \sigma_0 \alpha_0$ at initialisation), σ_0 the exploration scale, β the shaping coefficient and η_λ the controller step size. Everything else is as in Table 3.

Task	UTD	α_0	σ_0	β	η_λ
Ant	1.0	0.2	0.5	1.5	0.010
HalfCheetah	1.0	0.2	0.5	1.5	0.010
Hopper	0.6	0.2	0.5	0.8	0.002
Humanoid	1.0	0.2	0.1	2.4	0.120
Swimmer	0.2	0.2	0.5	0.025	0.100
Walker2d	0.2	0.2	0.5	1.5	0.010
CarCircle1	1.0	0.5	0.1	0.05	0.005
CarGoal1	1.0	0.5	0.1	0.05	0.055
PointCircle1	1.0	0.5	0.1	0.05	0.005
PointGoal1	0.25	0.5	0.1	0.10	0.005

H.3 BASELINES

All baselines use the same budget $d = 25$ and the same interaction budgets as Table 2, except RESPO (which enforces a zero-cost threshold by formulation). Each baseline is initialized from its published configuration; any adaptations or tuning are stated below.

H.3.1 SAC-LAGRANGIAN

Soft actor–critic with a Lagrangian multiplier updated by Adam on $\log \lambda$ from the realised episodic cost. Two backbones are used, matching the two task families (Table 5): the velocity family uses the standard SAC backbone; the Goal tasks use the slower actor of the reference safe-RL implementations, which we verified is *necessary* on these tasks (raising the actor learning rate from 5×10^{-6} to 3×10^{-4} collapses PointGoal1 to approximately zero return). The cost critic is floored at zero: because the actor minimises λQ_c without a lower bound, an unfloored cost critic diverges (we observed $Q_c \approx -800$ against a theoretical range of $[0, 100]$, with the run's return collapsing).

Table 5: SAC-Lagrangian. "Velocity" covers the six velocity tasks and the two Circle tasks; "Goal" covers CarGoal1 and PointGoal1.

Hyperparameter	Velocity / Circle	Goal
Critic / actor MLP	3×256, GELU	2×256, ReLU
Actor learning rate	3×10^{-4}	5×10^{-6}
Critic learning rate	3×10^{-4}	1×10^{-3}
Entropy temperature α	learned, $\alpha_0 = 0.2$	fixed 10^{-5}
α learning rate	3×10^{-4}	—
Policy delay	1	2
Gradient-norm clip	—	40
Multiplier learning rate	1×10^{-4}	1×10^{-5}
Multiplier initialisation	10^{-3}	0
Multiplier upper bound	100	100
Multiplier update	per completed episode	per gradient step after 100k steps
Cost window	1	10
Non-negative cost critic	yes	yes
Batch size / replay	256 / 10^6	256 / 10^6
UTD ratio	0.25	0.25
Warm-up transitions	30k	10k
Discounts γ, γ_c	0.99, 0.99	0.99, 0.99
Polyak τ	0.005	0.005

H.3.2 CAL

CAL (Wu et al., 2024) is initialized from its published configuration. Because the published navigation experiments use fixed layouts, we retune the SAC backbone for the randomized Circle tasks. Specifically, we change the actor learning rate from 5×10^{-6} to 3×10^{-4}, replace the fixed entropy temperature $\alpha = 10^{-5}$ with a learned temperature, increase the network depth from two to three hidden layers, change the policy delay from 2 to 1, and adjust the discounted per-step cost threshold from 1.0 to 0.5. CAL's safety-specific coefficients and multiplier update are otherwise unchanged. The Goal tasks retain the published backbone.

Common settings are a four-member cost-critic ensemble with ReLU activations, target updates every two gradient steps, multiplier initialization at zero, discounts $\gamma = \gamma_c = 0.99$ except $\gamma_c = 0.9$ on the Circle tasks, batch size 256, replay capacity 10^6, UTD ratio 0.25, and Polyak coefficient $\tau = 0.005$. Table 6 gives the per-task settings.

H.3.3 PPO-LAGRANGIAN

The on-policy reference is run at its published settings, except that we reduce the number of epochs for the navigation tasks so that their total interaction budget matches the 3M environment steps used by the other methods. No per-task hyperparameter tuning is applied. Table 7 lists the resulting

Table 6: CAL per-task settings. k is the cost-critic UCB coefficient, c the rectification coefficient. "done-mask" indicates that termination is masked in the cost backup; "$Q_c \geq 0$" that the cost critic is floored at zero.

Task	k	c	Multiplier LR	Internal d	Extra
Ant	0.0	0.01	1×10^{-5}	25	—
HalfCheetah	0.0	0.3	1×10^{-4}	25	$Q_c \geq 0$
Hopper	0.5	0.2	1×10^{-5}	25	done-mask, $Q_c \geq 0$
Humanoid	0.0	0.01	1×10^{-5}	25	done-mask
Swimmer	0.0	0.01	1×10^{-5}	25	—
Walker2d	0.0	0.2	1×10^{-4}	25	done-mask
CarCircle1	0.0	0.05	5×10^{-7}	25	retuned backbone, cost threshold 0.5
CarGoal1	0.0	0.3	5×10^{-7}	25	published backbone
PointCircle1	0.0	10.0	5×10^{-7}	25	retuned backbone, cost threshold 0.5
PointGoal1	0.0	10.0	5×10^{-7}	25	published backbone

configuration; rollout lengths follow the published protocol, while the navigation interaction budget differs only through this change in the number of epochs.

Table 7: PPO-Lagrangian.

Hyperparameter	Velocity	Humanoid	Navigation
Policy / value MLP	2×64, tanh	2×64, tanh	2×64, tanh
Learning rate	3×10^{-4}	3×10^{-4}	3×10^{-4}
Steps per epoch	2048	6000	20000
Epochs	500	500	150
Total environment steps	1.024M	3M	3M
Update iterations / epoch	40	40	40
Minibatch size	64	64	64
Clip ratio	0.2	0.2	0.2
Target KL (early stop)	0.02	0.02	0.02
γ / GAE λ	0.99 / 0.95	0.99 / 0.95	0.99 / 0.95
Cost GAE λ	0.95	0.95	0.95
Entropy coefficient	0	0	0
Gradient-norm clip	40	40	40
Critic ℓ_2 coefficient	10^{-3}	10^{-3}	10^{-3}
Multiplier learning rate	0.035	0.035	0.035
Multiplier initialisation	10^{-3}	10^{-3}	10^{-3}
Lagrange window	100 epochs	100 epochs	100 epochs

H.3.4 RESPO

We adapt the authors' released PyTorch RESPO implementation (Ganai et al., 2023) to Safety Gymnasium, preserving its policy objective, reachability targets, and multiplier update. The Gaussian actor and three value networks use two 256-unit tanh layers. Training uses four MPI workers, full-batch updates, and linearly decaying learning rates. We performed hyperparameter tuning on selected tasks, varying the actor learning rate, rollout size, target KL, and initial policy standard deviation. We additionally used observation normalization for Humanoid, which improves the performance compared to the default setting. The selected settings in Table 8 were then held fixed across the subsequent training seeds. All tasks use a 1,000-step episode limit and action repeat one; observation normalization is applied only to Humanoid.

We retain RESPO's released training constraint of zero. The common episodic cost threshold $d = 25$ is used only for reporting and assessing feasibility, not for its multiplier update. Learning curves show training-rollout return and cost.

Table 8: RESPO per-task settings. Budgets and rollout sizes count total environment transitions across four workers. Actor LR and σ_0 are the initial learning rate and policy standard deviation. Observation normalization is used only for Humanoid.

Task	Budget	Rollout	Actor LR	σ_0	Target KL
Ant	1M	10,000	6×10^{-4}	0.2	0.03
HalfCheetah	1M	10,000	6×10^{-4}	0.2	0.03
Hopper	1M	10,000	6×10^{-4}	0.2	0.03
Humanoid	3M	5,000	1×10^{-4}	0.2	0.03
Swimmer	1M	10,000	6×10^{-4}	0.2	0.03
Walker2d	1M	10,000	3×10^{-4}	0.2	0.03
CarCircle1	3M	30,000	1×10^{-4}	0.2	0.01
CarGoal1	3M	30,000	1×10^{-4}	$e^{-0.5}$	0.01
PointCircle1	3M	30,000	3×10^{-4}	$e^{-0.5}$	0.01
PointGoal1	3M	30,000	3×10^{-4}	$e^{-0.5}$	0.01

H.3.5 ALGD

We run ALGD (Cheng et al., 2026) from the authors' released PyTorch implementation (commit `1f6820b`), with the training harness adapted so that ALGD sees the same environments, horizons, budgets and evaluation as the other methods (Table 9). Table 10 lists the released hyperparameters, which we keep, and Table 11 the per-task settings.

Table 9: ALGD: released training protocol versus the protocol used here.

	Released code	This paper
Environments	Safety-Gym (`Safexp-*`), MuJoCo	Safety Gymnasium (all ten tasks)
Navigation layout	fixed for every episode and seed	resampled every episode
Episode horizon	400 (navigation)	1000
Goal reached	episode terminates	new goal spawned, episode continues
Cost budget	10 (navigation) / raw velocity	$d = 25$ on all tasks
Evaluation	1 episode	20 deterministic episodes

Cost threshold. ALGD constrains a discounted cost critic, and its released code converts an episodic budget into a per-step threshold $h = d\,(1 - \gamma_c^H)/\big((1 - \gamma_c)H\big)$. We apply this conversion on all ten tasks with $d = 25$, $\gamma_c = 0.99$ and H the number of decisions per episode, giving $h \approx 2.50$ for the Goal and velocity tasks ($H = 1000$) and $h \approx 9.19$ for the Circle tasks ($H = 250$ at action repeat 4).

H.4 Ablation configurations

The ablations of Figure 4 are one-factor changes to the corresponding constrained arm of Table 4; every other hyperparameter is held fixed.

- **Discounted critic.** Replaces the reachability critic with a standard discounted cost critic, keeping the same successor-state reward-shaping structure.
- **Cost-only shaping.** Shapes with the realised cost, $\tilde{r}_t = r_t - \beta\lambda c_t$, instead of the predicted reachability of the successor state.
- **Lag-style shaping.** Removes reward shaping entirely and puts the cost term into the candidate weights, i.e. the logits become $\hat{A}_i - \lambda \hat{A}_i^c$ with the cost advantage clipped symmetrically to $[-3, 3]$ to match the reward advantage's clip.

Gain matching. The shaping coefficient β deliberately *differs* between arms, because the quantity it multiplies differs in scale: a reachability value is a probability in $[0, 1]$, whereas a discounted cost value and a realised cost are not. Holding β fixed would therefore run the alternatives at a different

Table 10: ALGD hyperparameters (released defaults, shared by the DDPM and VE-SDE agents).

Group	Hyperparameter	Value
Networks	Reward critics (twin)	2×256, ReLU
	Cost-critic ensemble	4 members, 2×256, SiLU
	Score / noise network	3×128, ReLU, time embedding
	Denoising steps K	5
Diffusion	DDPM noise schedule	linear $\beta \in [10^{-4}, 0.02]$
	VE-SDE noise scale	geometric $\sigma \in [0.01, 1.0]$
Augmented Lagrangian	Penalty coefficient ρ	1.0
	Multiplier initialisation	$\lambda_0 = 0.6931$
	Multiplier update	Adam on $\log \lambda$, lr 3×10^{-4}
Score estimation	Monte-Carlo samples	4
	Score temperature β	1.0
Optimisation	Optimiser	Adam
	Actor / critic / cost-critic lr	3×10^{-4}
	Batch size / replay	256 / 10^6
	Discounts γ, γ_c	0.99, 0.99
	Polyak τ / target period	0.005 / every 2 updates
	Warm-up (uniform-random) steps	5k
	Updates per environment step	10 (default), see Table 11

Table 11: ALGD per-task settings. UTD is gradient updates per environment step.

Task	Agent	UTD
Ant	DDPM	10
HalfCheetah	DDPM	10
Hopper	DDPM	10
Humanoid	DDPM	10
Swimmer	DDPM	10
Walker2d	DDPM	10
CarCircle1	VE-SDE	10
CarGoal1	DDPM	10
PointCircle1	VE-SDE	10
PointGoal1	VE-SDE	10

penalty magnitude and measure "too much penalty" rather than "worse safety signal". We instead measured the realised magnitude of each shaping term on a matched run and set β so that the mean penalty matches RADPO's; Table 12 reports the resulting β.

H.5 Deterministic-evaluation curves

Figure 2 in the main text plots the on-policy training rollouts, which are the statistics the safety-multiplier controller regulates. Figure 7 plots the deterministic-evaluation counterpart over the same interaction budgets: 20 evaluation episodes at every checkpoint, with exploration noise disabled. The two views can differ appreciably in absolute cost, thus we report both rather than choosing one.

Table 12: Ablation arms. Every arm shares its task's η_λ, σ_0, α_0, UTD and interaction budget with the RADPO arm beside it, and all settings of Table 3.

Task	Arm	Cost critic	Shaping target	β
CarGoal1	RADPO	reachability	$V_{reach}(s')$	0.05
	discounted critic	discounted	$V_c(s')$	0.01
	cost-only shaping	-	c_t	0.05
	lag-style shaping	discounted	none (cost in weights)	—
PointCircle1	RADPO	reachability	$V_{reach}(s')$	0.05
	discounted critic	discounted	$V_c(s')$	0.01
	cost-only shaping	-	c_t	0.05
	lag-style shaping	discounted	none (cost in weights)	—
HalfCheetah	RADPO	reachability	$V_{reach}(s')$	1.5
	discounted critic	discounted	$V_c(s')$	0.254
	cost-only shaping	-	c_t	2.467
	lag-style shaping	discounted	none (cost in weights)	—

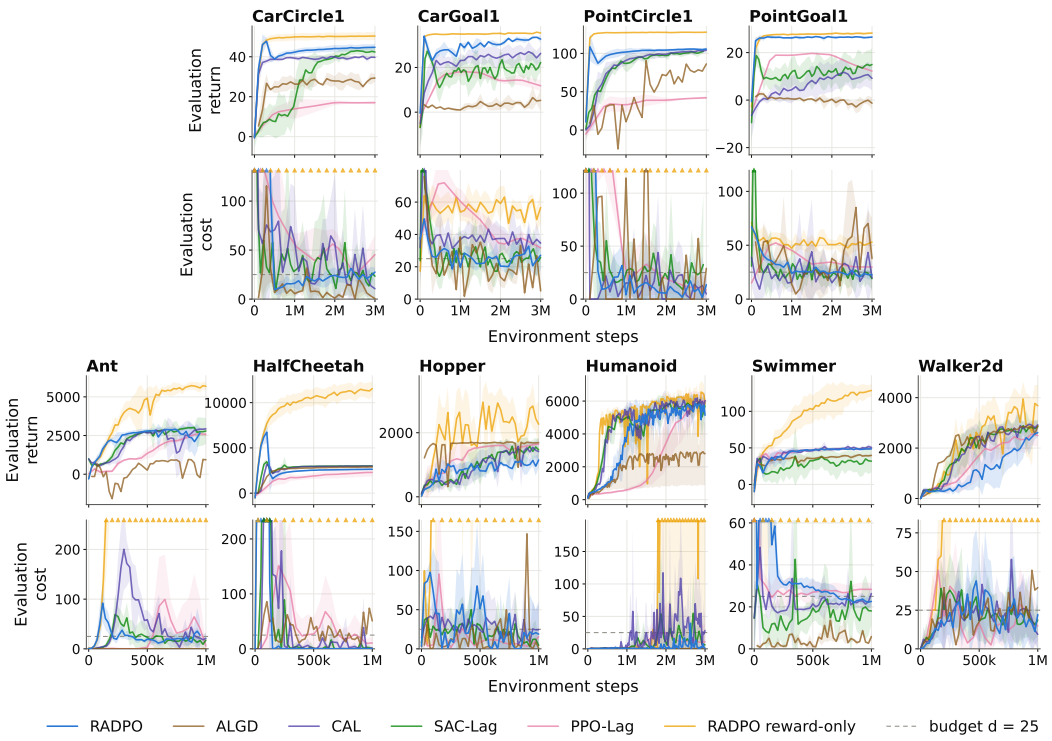

Figure 7: **Deterministic-evaluation return and cost across the ten tasks.** Evaluation return (upper row of each block) and evaluation cost (lower row), measured over 20 deterministic episodes per checkpoint; the training-rollout counterpart is Figure 2. Curves are means over the runs of an arm with one standard deviation shaded, and dashed horizontal lines mark the budget $d = 25$.