A VARIANCE-OPTIMAL REPLAY FRAMEWORK FOR SAFE OFF-POLICY REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing off-policy safe reinforcement learning methods inherit replay sampling strategies developed for unconstrained settings. Consequently, replay sampling remains optimized for reward rather than safety, compromising constraint satisfaction. Methods that prioritize transitions using safety-related heuristics reshape the sampling distribution without correction, inadvertently shifting the very solution that the agent is meant to reach. In this work, we treat the replay sampling distribution as a control variable to improve the agent's safety and learning performance. We prove that importance-weighted replay sampling preserves the *expected* constrained policy update, while affecting *only its variance*. We further show that reducing this variance lowers the probability of unsafe updates during training. Building on this result, we derive a closed-form *replay sampling strategy* that achieves the minimum possible variance across minibatches and reduces safety violations. At convergence, it asymptotically approaches the optimal policy of the underlying Constrained Markov Decision Process. Our sampling method is compatible with a wide range of gradient-based safe reinforcement learning algorithms and demonstrates consistent improvements over existing replay sampling methods on the Safety Gymnasium benchmark suite.

1 INTRODUCTION

Safe Reinforcement Learning (SRL) is increasingly being applied in safety-critical domains such as Cyber-Physical Systems, where unsafe actions can have catastrophic consequences. Such problems are commonly formulated as Constrained Markov Decision Processes (CMDPs), in which the agent maximizes reward while keeping the expected safety cost below a fixed budget (Achiam et al., 2017). A scalable approach to solve the optimization problems formulated in CMDPs is via actor-critic methods with off-policy learning that use minibatches sampled from a *replay buffer* for updating the policy (Haarnoja et al., 2018; Stooke et al., 2020). Off-policy updates are calculated from the sampled transitions from the replay buffer, where gradients with respect to reward and cost gradually change the policy parameters. Consequently, the replay buffer's sampling distribution directly affects the estimated gradients and, in turn, the policy updates and overall learning trajectory (Fedus et al., 2020).

The reward and cost gradients represent two competing objectives. The reward gradient aims to improve performance, while the cost gradient aims to reduce safety violations. Since these gradients can point in different directions (Gu et al., 2024), the policy update must appropriately balance their contributions to avoid improving one objective at the expense of the other. Existing on-policy safe RL methods address this gradient conflict in different ways. Lagrangian methods combine the two gradients using a learned penalty (Stooke et al., 2020), Primal methods alternate between reward and cost updates (Xu et al., 2021), and Projection-based methods modify the gradients to satisfy the safety constraints (Gu et al., 2024). However, these methods estimate the gradients from a small set of sampled trajectories in their rollout buffers, which can result in fidgety gradient estimates each time the buffer is updated. An unfavorable set of samples may distort the reward or cost gradient, leading to an unsafe policy update with constraint violations. This sampling variability can lead to high estimation variance.

Off-policy methods can reduce this estimation error by reusing transitions collected over many interactions through an intelligent sampling strategy in the replay buffer (Fedus et al., 2020; Amirabadi

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.

et al., 2026). However, existing off-policy safe RL methods do not explicitly address the reward-cost gradient conflict while sampling from the replay buffer. Their replay prioritization strategies are largely heuristic (Mnih et al., 2015; Tan & Huang, 2024; Hassani et al., 2025) and reward-centric (Andrychowicz et al., 2017; Wang et al., 2025), without explicitly considering safety as a criterion. These strategies reshape the sampling distribution without correcting for the resulting bias. In addition, replay buffers contain transitions generated by a mixture of past behavior policies, so both the visited states and logged actions can differ from those that the current policy would generate (Liu et al., 2019). Existing off-policy methods do not explicitly correct for these state and action distribution mismatches when prioritizing and sampling replay data. As a result, the sampled transitions may not provide unbiased estimates of the current policy's reward and cost gradients, potentially shifting the policy update away from the desired CMDP solution.

These limitations highlight a fundamental gap in existing safe RL methods. On-policy methods explicitly address the reward-cost gradient conflict but suffer from noisy gradient estimates due to the limited number of sampled trajectories. In contrast, off-policy methods provide a large replay buffer that can improve gradient estimation. However, existing methods do not adequately address gradient conflict, the state distribution mismatch, bias introduced by replay sampling, and safety when prioritizing replay buffer elements. This motivates the following research question: *How should the replay sampling distribution be designed to simultaneously provide accurate reward and cost gradient estimates, resolve their conflict, and ensure convergence toward the desired CMDP solution?* In this work, we try to address this research question with the following contributions.

1. We derive, in closed form, a replay sampling strategy that minimizes the variance of the reward and cost gradients and, in turn, the variance of the constrained policy update. We prove that it achieves the lowest variance attainable by any sampler.
2. We show that a specialized importance weight leaves the expected constrained policy update unchanged, so replay prioritization can never shift the CMDP solution of any optimizer built on the reward and cost gradients. Consequently, the sampler influences only the variance of the update and never its converged solution, ensuring that the improved gradient estimation comes at no cost to convergence.
3. We show that combining the proposed sampler with any gradient-based safe optimizer yields a safe RL algorithm that preserves the optimizer's constrained solution while reducing its per-step violation probability. We validate this by integrating the sampler with several representative safe optimizers across Safety Gymnasium environments, demonstrating its effectiveness through extensive experiments.

2 Related Work

Prioritized experience replay. The most common way to shape the replay sampling distribution is Prioritized Experience Replay (PER), which samples transitions in proportion to assigned priorities, most often the Temporal-Difference (TD) error (Schaul et al., 2015). However, TD error only indicates a transition's usefulness and tends to oversample *noisy* transitions that aid neither reward maximization nor safety (Pan et al., 2022; Sekar et al., 2020). This can inflate the variance of the reward and cost gradient estimates because different minibatches can produce substantially different gradient estimates (Katharopoulos & Fleuret, 2018). Later metrics such as regret (Liu et al., 2021), reducible loss (Sujit et al., 2023), and uncertainty (Carrasco-Davis et al., 2025) improve reward learning but remain reward-centric and ignore which transitions best inform the safety constraint. Fan et al. (2025) prioritizes using the TD error of a learned safety estimate, but this heuristic is tied to their value-based masking method and gives no guarantee that the selected transitions improve constraint satisfaction. Unlike these existing prioritization rules, we treat the replay sampler as an importance-corrected control variable that accounts for both reward and safety. The importance correction ensures that prioritization changes only the variance of the policy update without changing its expectation. We then design the sampling distribution to minimize this variance.

Safe reinforcement learning methods. Safe RL is commonly cast as a CMDP (Achiam et al., 2017), and a large family of methods resolves the reward-cost conflict at the gradient level. Lagrangian and primal–dual methods combine the two gradients through a learned multiplier (Stooke et al., 2020; Haarnoja et al., 2018; Yang et al., 2021), primal methods alternate between the reward and cost gradients by feasibility (Xu et al., 2021), and projection-based methods manipulate them

geometrically (Yu et al., 2020; Gu et al., 2024). Others enforce safety through penalties, barriers, or edits to the policy output (Dai et al., 2023; Cheng et al., 2019; Yu et al., 2022; Liu et al., 2022; Yao et al., 2023; Honari et al., 2024; Hazra et al., 2025). All of these still estimate the gradients from sampled data and inherit its variance. Rather than proposing a new gradient optimizer, we remain agnostic to the choice of optimizer. We provide any gradient-based optimizer with lower-variance estimates through the replay sampler and prove that this preserves its constrained solution.

State-visitation correction. Off-policy learning involves a mismatch between the distribution of stored data collected by past behavior policies and the data distribution of the current policy. Standard importance sampling corrects for the mismatch in actions (Munos et al., 2016). However, it does not correct the mismatch in state visitation, which can bias the policy gradient. The works in Liu et al. (2019); Nachum et al. (2019); Zhang et al. (2020); Hallak & Mannor (2017) address this issue by correcting the state distribution using logged data, following the approach of Liu et al. (2018). We use state-visitation correction as a standard component to obtain an unbiased replay estimator for the current policy. This allows our sampler to reduce the variance of the policy update without changing its expected value.

3 PRELIMINARIES

We consider a Constrained Markov Decision Process (CMDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, P, r, c, \gamma, \rho_0)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, P is the transition kernel, r is the reward function, c is the cost function, $\gamma \in (0, 1)$ is the discount factor, and ρ_0 is the initial state distribution. At time t, the agent is in state $s_t \in \mathcal{S}$, takes action $a_t \in \mathcal{A}$, and receives reward $r(s_t, a_t)$ and cost $c(s_t, a_t)$. For a policy π_θ parameterized by θ, the discounted reward and cost returns are $J_r(\pi_\theta) = \mathbb{E}_{\pi_\theta}[\sum_t \gamma^t r(s_t, a_t)]$ and $J_c(\pi_\theta) = \mathbb{E}_{\pi_\theta}[\sum_t \gamma^t c(s_t, a_t)]$. The agent maximizes J_r subject to $J_c \leq d$, where d is the cost budget. With a non-negative multiplier $\lambda \geq 0$, the Lagrangian is $L(\theta, \lambda) = J_r - \lambda(J_c - d)$. The constrained optimum satisfies the four Karush–Kuhn–Tucker (KKT) (Achiam et al., 2017). conditions. These conditions are stationarity: $\nabla_\theta J_r - \lambda^\star \nabla_\theta J_c = 0$, feasibility: $J_c \leq d$, non-negativity: $\lambda^\star \geq 0$, and complementary slackness: $\lambda^\star(J_c - d) = 0$. This point is the target that a safe learner aims to preserve.

Reward and cost gradients. The KKT stationarity condition depends on the reward and cost gradients, which determine how the policy parameters should change to improve reward and cost, respectively. The reward gradient $g_r = \nabla_\theta J_r$ points in the direction of fastest reward improvement, as shown by the blue arrow in Fig. 1(a), while the cost gradient $g_c = \nabla_\theta J_c$ points in the direction of fastest cost increase, shown by the red arrow in Fig. 1(a). Using the policy gradient theorem (Sutton et al., 2000), these gradients can be estimated from sampled transitions as $g_r = \mathbb{E}_{d^\pi}\big[Q_r(s, a)\, \phi_\theta(s, a)\big]$ and $g_c = \mathbb{E}_{d^\pi}\big[Q_c(s, a)\, \phi_\theta(s, a)\big]$. Here, $\phi_\theta(s, a) = \nabla_\theta \log \pi_\theta(a \mid s)$ is the score function, which describes how the policy changes the probability of taking action a in state s, while Q_r and Q_c are the reward and cost action-value functions of the current policy π_θ, defined as $Q_r(s, a) = \mathbb{E}_{\pi_\theta}\big[\sum_{t\geq 0} \gamma^t r(s_t, a_t) \mid s_0 = s,\, a_0 = a\big]$ and $Q_c(s, a) = \mathbb{E}_{\pi_\theta}\big[\sum_{t\geq 0} \gamma^t c(s_t, a_t) \mid s_0 = s,\, a_0 = a\big]$. In practice, the gradients are estimated from a minibatch of stored transitions $\tau_i = (s_i, a_i, r_i, c_i, s_i')$. For each transition τ_i, the corresponding single-sample estimates of the reward and cost gradients are $u_i = Q_r(s_i, a_i)\, \phi_\theta(s_i, a_i)$ and $v_i = Q_c(s_i, a_i)\, \phi_\theta(s_i, a_i)$, respectively. Reward improvement and cost reduction can require different policy changes, causing the two gradients to conflict. Their conflict is measured by the inner product $\omega = \langle g_r,\, g_c \rangle = \|g_r\|\, \|g_c\| \cos\vartheta$. Here, ϑ is the angle between g_r and g_c (shown in Fig. 1(a)), and a positive value $\omega > 0$ indicates that moving in the reward-improving direction also increases cost. The conflicting component of the reward gradient, shown by the grey arrow in Fig. 1(b) can be removed by projecting g_r onto the direction orthogonal to g_c, shown by the green arrow in Fig. 1(c). This gives the conflict-free reward direction, g_r^+, as shown below.

$$g_r^+ = g_r - \frac{\omega}{\|g_c\|^2}\, g_c, \qquad \langle g_r^+,\, g_c \rangle = 0. \tag{1}$$

The term g_r^+ removes the component of the reward gradient that increases the cost, so moving along g_r^+ improves the reward while keeping the cost unchanged (Yu et al., 2020).

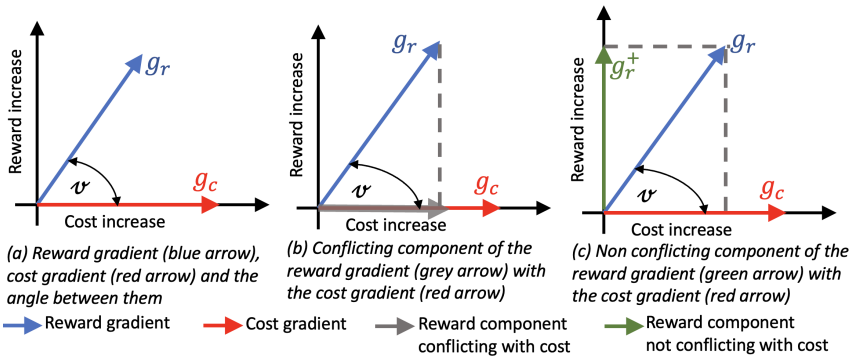


Figure 1: Visualization of the reward and cost gradients, along with the reward components that conflict and do not conflict with the cost gradient.

Gradient-based optimizer. A gradient-based policy optimizer determines the policy update by linearly combining the reward and cost gradients. This update is represented as $\Delta = \mathcal{F}(g_r, g_c)$, where $\mathcal{F}$ specifies how the two gradients are combined. Different safe RL methods use different forms of $\mathcal{F}$. For example, Lagrangian methods use $\mathcal{F}(g_r, g_c) = g_r - \lambda g_c$ (Stooke et al., 2020), constrained reward policy optimization based methods selects either $\mathcal{F}(g_r, g_c) = g_r$ or $\mathcal{F}(g_r, g_c) = -g_c$ based on constraint satisfaction (Xu et al., 2021), and projection-based methods use $\mathcal{F}(g_r, g_c) = g_r^+$ (Yu et al., 2020; Gu et al., 2024). These different rules divide the gradient space into regions in which the optimizer follows a specific update rule. Notice that the update rules at any region are linear combinations of g_r and g_c for a fixed ϑ. Ideally, with exact gradient values, the resulting update would have been $\Delta^\star = \mathcal{F}(g_r, g_c)$. In practice, the gradients are estimated from a minibatch, resulting in $\Delta_q = \mathcal{F}(\hat{g}_r, \hat{g}_c)$. The difference between $\Delta^\star$ and Δ_q represents the error in the policy update caused by gradient estimation.

Replay corrections. Off-policy learning uses data collected by past behavior policies rather than the current policy, creating two distribution mismatches. The first is an action mismatch because the stored actions were generated by a behavior policy π_{beh} rather than the current policy π_θ. It is corrected using the policy likelihood ratio $\rho_i = \pi_\theta(a_i \mid s_i)/\pi_{\text{beh}}(a_i \mid s_i)$ Shelton (2001). The second is a state-visitation mismatch because the states in the replay buffer follow the behavior visitation distribution d^{beh} rather than the current discounted visitation distribution d^{π_θ}. It is corrected using $\beta(s_i) = d^{\pi_\theta}(s_i)/d^{\text{beh}}(s_i)$, which can be estimated from buffer samples using the density-ratio method of Liu et al. (2019; 2018). The two corrections give the importance weight as shown below.

$$w_i = \beta(s_i)\rho_i = \frac{d^{\pi_\theta}(s_i)}{d^{\text{beh}}(s_i)} \frac{\pi_\theta(a_i \mid s_i)}{\pi_{\text{beh}}(a_i \mid s_i)}. \tag{2}$$

To prevent a few samples with very large weights from dominating the gradient estimate, the importance weight is clipped to $[1/C, C]$ for a fixed $C \geq 1$, as: $w_i = \text{clip}\big(\beta(s_i)\rho_i,\ 1/C,\ C\big) = \min\Big\{\max\big\{\beta(s_i)\rho_i,\ 1/C\big\},\ C\Big\}$.

4 Methodology

Our methodology proceeds in four stages. *First*, we construct an importance-corrected replay estimator that preserves the expected update of a gradient-based safe RL optimizer, ensuring that the sampling distribution does not alter its constrained solution. *Second*, we derive a closed-form sampling distribution that minimizes the variance of the reward and cost gradient estimates. *Third*, we bound the deviation of the resulting policy update from the ideal update and relate it to the probability of selecting an incorrect optimizer branch due to gradient estimation error. Finally, we show that the proposed replay-sampling method can be integrated with different gradient-based safe RL optimizers while reducing the probability of safety violations during training. In the next sections, we will discuss these stages.

4.1 UNBIASED OFF-POLICY REPLAY

Let $b(i)$ denote the baseline distribution over transitions in the replay buffer and $q(i)$ the distribution used for sampling. After sampling a transition $i \sim q$, the importance-corrected reward and cost contributions are shown below.

$$X_i = \frac{b(i)}{q(i)} w_i u_i, \qquad Y_i = \frac{b(i)}{q(i)} w_i v_i. \tag{3}$$

The ratio $b(i)/q(i)$ corrects the change from the baseline distribution b to the sampling distribution q, while $w_i = \beta(s_i)\rho_i$ corrects the state-visitation and action mismatches between the replay data and the current policy (Liu et al., 2019). Therefore, for any sampler q with $q(i) > 0$ whenever $b(i) > 0$, the corrected estimates preserve the original expected reward and cost gradients.

Lemma 1. *For any replay buffer sampling distribution q with $q(i) > 0$ whenever the baseline distribution $b(i) > 0$, $\mathbb{E}_{i\sim q}[X_i] = g_r$ and $\mathbb{E}_{i\sim q}[Y_i] = g_c$. Hence, $\mathbb{E}_{i\sim q}[X_i - \lambda Y_i] = g_r - \lambda g_c$ for every sampler, and the mean gradients g_r and g_c used by any optimizer $\mathcal{F}$ are same for every choice of q.*

Proof: Provided in the supplementary material.

From the proof, the factor $b(i)/q(i)$ removes the bias introduced by sampling transitions according to q. Therefore, changing q does not change the expected update $\mathcal{F}(g_r, g_c)$ or the optimizer's constrained solution. With the expected update fixed, the choice of q affects only the variance of the gradient estimates, which we minimize next.

4.2 THE VARIANCE-MINIMIZING SAMPLER

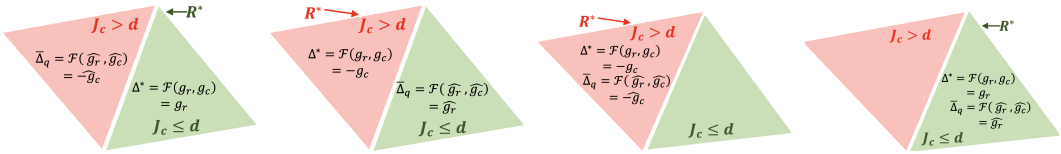
Importance correction in Lemma 1 ensures that the expected minibatch estimates of reward and cost gradients remain equal to their corresponding full-replay-buffer estimates. In this section, we propose a variance-minimising sampler that reduces the variance of the reward and cost gradients, thereby avoiding conflict between them during policy update. In a minibatch the per-transition cost-gradient estimate v_i is projected onto the unit cost-gradient direction $\tilde{g}_c = g_c/\|g_c\|$, along which the safety cost rises. Similarly, the per-transition reward-gradient estimate u_i is projected onto the unit conflict-free reward direction $\tilde{g}_r^+ = g_r^+/\|g_r^+\|$, along which reward improves without affecting cost. To quantify how aligned the per-transition estimates of cost and reward gradients are with their minibatch-wise estimates, we use the scalar scores $s_i^c = w_i\langle v_i,\, \tilde{g}_c\rangle$ and $s_i^r = w_i\langle u_i,\, \tilde{g}_r^+\rangle$, respectively ($\langle\cdot,\,\cdot\rangle$ denotes the inner (dot) product). Importance-corrected estimations of these scalar scores are $Z^c = \frac{b(i)}{q(i)} s_i^c$ and $Z^r = \frac{b(i)}{q(i)} s_i^r$. Using Lemma 1, it can be argued that $\mathbb{E}_q[Z^c] = \|g_c\|$ and $\mathbb{E}_q[Z^r] = \|g_r^+\|$ (proof provided in supplementary material, referred as Proposition 1). An incorrect step is the event $Z^c \le 0$, which denotes the single minibatch estimate and the true estimate of the cost-gradient $\tilde{g}_c$ point in conflicting directions. Therefore, an update that should move against g_c and reduce cost on average instead increases it. Cantelli's inequality (Boucheron et al., 2013) bounds this probability of conflict (i.e., $Z^c \le 0$) as shown below.

$$P(Z^c \le 0) \le \frac{\sigma_c^2}{\sigma_c^2 + \|g_c\|^2} \tag{4}$$

Here, $\sigma_c^2 = \mathrm{Var}(Z^c)$ is the variance of Z^c across minibatches drawn by the sampler q. The mean $\|g_c\|$ is fixed, so this bound tightens only by shrinking σ_c^2, which the sampler can control. Similarly, the bound on Z^r can be tightened by minimizing the variance $\sigma_r^2 = \mathrm{Var}(Z^r)$. Here, the event $Z^r \le 0$ means that a minibatch estimation of the reward gradient opposes the true reward-progress direction. To minimise the probability of conflicts between the minibatch and per-transition gradient estimates, we minimise a weighted sum of the variances of Z^c and Z^r, which are computed as $\sigma_c^2 = \sum_i \frac{b(i)^2}{q(i)}(s_i^c)^2 - \|g_c\|^2$ and $\sigma_r^2 = \sum_i \frac{b(i)^2}{q(i)}(s_i^r)^2 - \|g_r^+\|^2$, respectively. Considering a single weight parameter $\kappa \ge 0$, the optimisation problem inside a *trust region* is as shown below.

$$\min_q \; \sigma_c^2 + \kappa\, \sigma_r^2 \quad \text{under the } \textit{constraint}: \quad D_{\mathrm{KL}}(q \,\|\, b) \le \epsilon \tag{5}$$

Here, $\epsilon > 0$ is the *trust-region radius* that limits how far the sampler q may move from the uniform buffer distribution b. The optimization problem can be reduced to minimizing $\sum_i A_i/q(i)$, where $A_i = b(i)^2\big[(s_i^c)^2 + \kappa(s_i^r)^2\big]$. The following theorem establishes a closed-form solution of the variance-optimal sampling distribution that minimizes $\sum_i A_i/q(i)$.



(a) $\Delta^\star$, $\bar{\Delta}_q$ fall in different regions (wrong branch). (b) $\Delta^\star$, $\bar{\Delta}_q$ fall in different regions (wrong branch). (c) $\Delta^\star$, $\bar{\Delta}_q$ fall in same regions (correct branch). (d) $\Delta^\star$, $\bar{\Delta}_q$ fall in same regions (correct branch).

Figure 2: Ideal step $\Delta^\star$ and minibatch-estimated step Δ_q under a CRPO-based (Xu et al., 2021) optimizer, shown across the safe region $J_c \le d$ (green coloured) and the violated region $J_c > d$ (red coloured).

Theorem 1. *Over all admissible samplers, the objective in Eq.* (5) *(without the trust region constraint) is minimized uniquely by the optimal sampler formulated as:* $q^\star(i) = \frac{b(i)\sqrt{(s_i^c)^2+\kappa(s_i^r)^2}}{\sum_j b(j)\sqrt{(s_j^c)^2+\kappa(s_j^r)^2}}$, *which attains the variance floor* $\left(\sum_i \sqrt{A_i}\right)^2 - \|g_c\|^2 - \kappa\|g_r^+\|^2$.

Proof: Provided in the supplementary material.

The optimal sampler formulation, subject to the trust-region constraint, changes as follows.

$$q_\epsilon^\star(i) = \frac{b(i)\exp(\rho_i/\eta)}{\sum_j b(j)\exp(\rho_j/\eta)}, \tag{6}$$

This is formally proved in Proposition 2 (provided in the supplementary material). Here, $\rho_i = \sqrt{(s_i^c)^2 + \kappa(s_i^r)^2}$. The temperature parameter η controls how much the sampling is spread across transitions. A large η gives the baseline distribution b, while a small η gives the minimum-variance sampling distribution. In the following section, we demonstrate how variance-optimal sampling bounds the deviation of the policy update from the ideal update for any gradient-based optimizer.

4.3 Effect of the Sampler on the Policy Update

A minibatch of size n drawn under sampler q returns unbiased estimates $\hat{g}_r$ and $\hat{g}_c$, and the resulting update step is $\Delta_q = \mathcal{F}(\hat{g}_r, \hat{g}_c)$. As explained in Sec. 3, $\mathcal{F}$ is a gradient optimizer and it can be expressed as a linear combination of $\hat{g}_r$ and $\hat{g}_c$ (for a fixed ϑ), i.e., $\mathcal{F} = A_k\hat{g}_r + B_k\hat{g}_c$ at a region R_k in the gradient estimation space. Therefore, it is linear on the region $R^\star$ containing the correct update rule. The ideal policy gradient update step in $R^\star$ is denoted as $\Delta^\star = \mathcal{F}(g_r, g_c)$. The effective policy gradient update step $\bar{\Delta}(q) = \mathbb{E}_q[\Delta_q]$ is computed as the weighted average of the update steps across the minibatch sampled using q.

The only source of sampling error is a wrong branch in which $(\hat{g}_r, \hat{g}_c)$ lies in a region other than $R^\star$, while (g_r, g_c) lies in $R^\star$, as shown in Fig. 2. The green area represents the safe region $J_c \le d$, whereas the red area represents the violated region $J_c > d$. Figures 2a–2d illustrate the four possible cases in which $\bar{\Delta}_q$ and $\Delta^\star$ either lie in the same region or in different regions, together with the corresponding CRPO (Xu et al., 2021) update rule in each case. We measure this error using the two metrics Z^r and Z^c.

Theorem 2. *Let $\mathcal{F}$ be a gradient-based optimizer with constant $L_\mathcal{F} = \max_k(\|A_k\| + \|B_k\|)$, and suppose the clipped weights in Eq. 7 ensure $\|\hat{g}_r\|, \|\hat{g}_c\| \le \bar{G}$, where $\bar{G} > 0$ is an upper bound on the norm of the estimated gradients $\hat{g}_r$ and $\hat{g}_c$. Let $P_{\text{flip}}(q)$ denote the probability that the estimated gradients leave the region $R^\star$. Then $\left\|\bar{\Delta}(q) - \Delta^\star\right\| \le 2L_\mathcal{F}\bar{G}\,P_{\text{flip}}(q)$, where* $P_{\text{flip}}(q) \le \frac{\sigma_c^2(q)}{\sigma_c^2(q)+\|g_c\|^2} + \frac{\sigma_r^2(q)}{\sigma_r^2(q)+\|g_r^+\|^2}$.

Proof. Provided in the supplementary material.

As the variance-optimal sampler $q^\star$ in Theorem 1 minimises the variances σ_c^2 and σ_r^2 optimally, $P_{\text{flip}}(q)$ is bounded with a stricter upper limit compared to other samplers.

Corollary 1. *Let $a = q_\epsilon^\star$ and let b be any sampler with $\sigma_c^2(a) \le \sigma_c^2(b)$ and $\sigma_r^2(a) \le \sigma_r^2(b)$. Then,* $\langle\Delta^\star, \bar{\Delta}(a)\rangle \ge \|\Delta^\star\|^2 - 2L_\mathcal{F}\bar{G}\,\|\Delta^\star\|\,P_{\text{flip}}(b)$.

Proof. Provided in the supplementary material.

Therefore, among all samplers, the effective policy update step that our variance-optimal sampler feeds any gradient-based policy optimiser carries the largest guaranteed alignment with the ideal policy update step. Its certified lower bound on this alignment is at least as large as that of any baseline sampler with larger variances.

4.4 Algorithmic Implementation

We compile the discussions presented so far into Algorithm 1, which describes our proposed method named *Variance-Optimal Sampler for Replay* (VOSR).

Algorithm 1 Variance Optimal Sampler for Replay (VOSR) training step

Input replay buffer $\mathcal{B}$, baseline sampler b, temperature parameter η, weight parameter κ, gradient-based policy optimizer $\mathcal{F}$, state-visitation ratio estimator β
Output updated policy parameter θ for actor network

1: Refresh the state-visitation ratio $\beta(\cdot) = d^{\pi}/d^{\text{beh}}$ and set $w_i = \beta(s_i)\,\pi_\theta(a_i \mid s_i)/\pi_{\text{beh}}(a_i \mid s_i)$, clipped to $[1/C, C]$
2: Estimate g_r, g_c using Eq. (3) on a candidate minibatch from $\mathcal{B}$
3: Compute $\tilde{g}_c = g_c/\|g_c\|$ and $\tilde{g}_r^+ = g_r^+/\|g_r^+\|$ using Eq. (1)
4: Compute scores s_i^c, s_i^r and $\rho_i = \sqrt{(s_i^c)^2 + \kappa(s_i^r)^2}$
5: Build sampler $q_\epsilon^\star(i) \propto b(i)\exp(\rho_i/\eta)$ from Eq. 6 and prepare minibatch $M \sim q_\epsilon^\star$
6: Update both critics using $\frac{b(i)}{q_\epsilon^\star(i)} w_i h(\tau_i)$ ▶ $h(\tau_i)$ = *gradient estimate of critic's* τ_i *transition*
7: Re-estimate $\hat{g}_r, \hat{g}_c$ from the minibatch M
8: Set $\Delta \leftarrow \mathcal{F}(\hat{g}_r, \hat{g}_c)$ with any gradient-based optimizer
9: Update policy parameter: $\theta \leftarrow \theta + \alpha\Delta$
10: **return** Updated θ for actor network

At each training step, the algorithm refreshes the state-visitation correction ratio, estimates the reward and cost gradients, computes the scores, updates both critics using the importance-sampled targets, and finally passes the precise gradient estimates to a gradient-based optimiser $\mathcal{F}$. Since Algorithm 1 changes only the replay sampling and gradient estimation procedure, the update rule $\mathcal{F}$ remains completely unchanged and can be chosen by the practitioner. In this work, we consider three such gradient-based optimizers, namely the Lagrangian-based CMDP optimizer (Stooke et al., 2020), CRPO (Xu et al., 2021), and PCRPO (Gu et al., 2024). The following theorem proves that the resulting method is a safe reinforcement learning algorithm that converges to the stationary points of a CMDP.

Theorem 3. *Let $\mathcal{F}$ be a safe gradient optimizer that converges to a KKT point of the CMDP when run with exact gradients. Wrapping $\mathcal{F}$ with the unbiased variance-optimal sampler as implemented in Algorithm 1, produces an algorithm with the same constrained stationary set as $\mathcal{F}$.*

Proof. The proof is provided in the supplementary material.

5 Experiments

In this section, we evaluate the proposed VOSR method on standard Safe RL benchmark environments. We first compare its reward and cost performance against state-of-the-art replay sampling strategies. We then evaluate its ability to reduce gradient estimation variance and finally assess its safety by measuring the frequency of constraint violations during training.

Setup: We evaluate VOSR on Safety Gymnasium benchmarks (Ray et al., 2019; Ji et al., 2023), covering velocity locomotion tasks (Ant Velocity, Half Cheetah Velocity, Humanoid Velocity, Swimmer Velocity, and Walker 2d Velocity) and navigation tasks (Point Goal, Point Push, Point Button 1, Car Button 1, and Car Goal 1). Since VOSR is optimizer-agnostic, we integrate it with three gradient-based safe RL optimizers: SAC-Lagrangian (Haarnoja et al., 2018), CRPO (Xu et al., 2021), and PCRPO (Gu et al., 2024). For each optimizer, we compare VOSR with four replay sampling strategies while keeping the optimizer unchanged, namely Uniform Replay, TD-PER (Schaul et al., 2015),

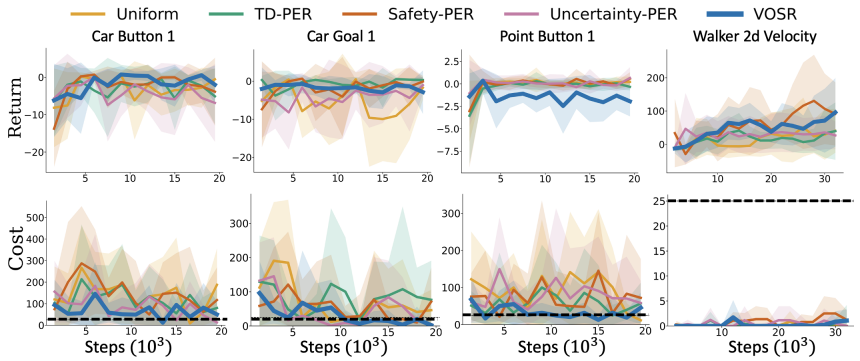


Figure 3: Comparison of VOSR (blue) and state-of-the-art replay sampling methods using **SAC-Lagrangian** policy optimization on showcase environments. The top row shows the mean episode return and the bottom row the mean episode cost.

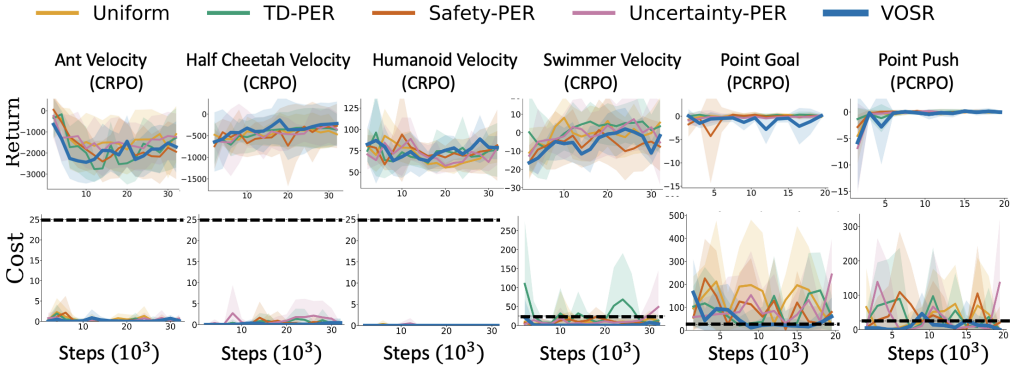


Figure 4: Comparison of VOSR (blue) and state-of-the-art replay sampling methods using **CRPO** (Xu et al., 2021), and **PCRPO** (Gu et al., 2024) policy optimization on showcase environments. The top row shows the mean episode return, and the bottom row the mean episode cost.

Safety-PER (Fan et al., 2025), and Uncertainty-PER (Carrasco-Davis et al., 2025). Every configuration is trained with seven random seeds under a fixed cost budget. The learning and cost curves report the mean and one standard deviation across the seven seeds, while the cost plots additionally include a dashed line indicating the cost budget. Hardware specifications and hyperparameter settings are provided in the supplementary material.

Return and cost across optimizers: Figures 3, and 4 report the training performance in terms of return and cost on representative showcase environments for SAC-Lagrangian, CRPO, and PCRPO respectively. In each figure, the top row shows the return, where higher is better, and the bottom row shows the cost, which should remain below the safety budget of 25. This value is the default safety threshold for Safety Gymnasium environments and is represented by the black dashed lines in the figures. Each column corresponds to one environment, each colour denotes a replay sampling strategy, and the bold blue curve represents VOSR. Across the locomotion tasks (SafeAntVelocity, SafeHalfCheetahVelocity, SafeHumanoidVelocity, and SafeSwimmerVelocity, and SafeWalker2dVelocity), VOSR drives the cost close to zero and well under the budget throughout training while matching or exceeding the return of the competing samplers. This shows that the improved safety does not come at the expense of task performance. The same trend is observed on the more challenging navigation tasks, SafeCarButton1, SafeCarGoal1, SafePointButton1, SafePointGoal2, and SafePointPush1. Although the returns of all methods are relatively close and often overlap, their costs differ substantially. VOSR keeps the cost near or below the budget, whereas Uniform Replay, TD-PER, Safety-PER, and Uncertainty-PER exhibit repeated cost spikes well above the safety limit (bottom rows of Figs. 3 and 4). These results agree with Theorem 2 and Corollary 1, which predict that reducing the gradient estimation variance keeps the optimizer's realized update closer to its ideal update, thereby improving safety without sacrificing return.

Variance of the safety estimate: Table 1 compares the variance of the cost gradient produced by different replay sampling strategies. For each environment, we compute the cost-gradient vari-

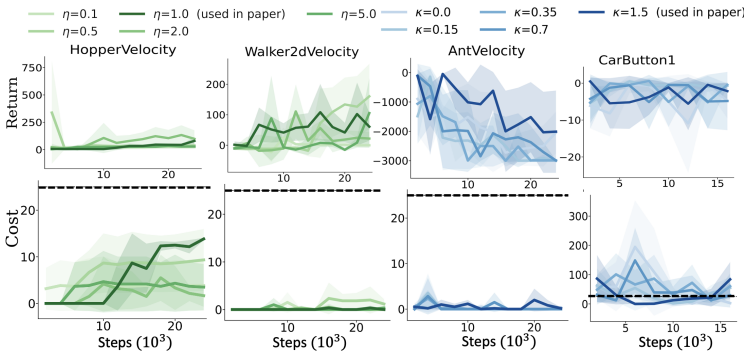


Figure 5: Effect of the parameters η and κ on the performance of VOSR with SAC-Lagrangian.

ance, average it over the three optimizers and seven random seeds, and normalize it by the variance of Uniform Replay. Therefore, Uniform Replay has a normalized variance of 1.00, and lower values indicate a more accurate and less noisy gradient estimate. TD-PER, Safety-PER, and Uncertainty-PER all produce normalized variances greater than 1.00, indicating that they increase the noise in the cost gradient compared with Uniform Replay. In contrast, VOSR achieves a normalized variance below 1.00 in every environment, with an average value of 0.57. This means that VOSR reduces the cost-gradient variance by about 43% compared with Uniform Replay.

Cost-violation fraction: The last row of Table 1 reports the cost-violation fraction, defined as the total number of training steps at which the cost exceeds the budget divided by the total number of training steps. The reported values are averaged over all environments, random seeds, and gradient-based optimizers. VOSR achieves the lowest violation fraction of 0.080, compared with 0.103 for TD-PER, 0.108 for Safety-PER, and 0.107 for Uncertainty-PER. Thus, VOSR violates the safety budget about 22% less often than TD-PER and about 26% less often than Safety-PER. These results are consistent with Theorem 3.

Table 1: Cost-gradient variance relative to Uniform replay. Lower is better.

Environment	TD-PER	Safety-PER	Unc.-PER	VOSR
Ant	2.43	2.14	1.44	**0.80**
HalfCheetah	7.28	2.01	1.65	**0.85**
Hopper	2.15	1.77	1.26	**0.28**
Humanoid	2.41	3.06	1.67	**0.99**
Swimmer	2.51	2.90	1.69	**0.40**
Walker2d	2.12	1.25	1.06	**0.52**
Button1	1.35	1.80	1.25	**0.38**
Push1	1.43	1.45	1.22	**0.35**
Average	2.71	2.05	1.41	**0.57**
Violation	0.103	0.108	0.107	**0.080**

Ablation studies on η and κ: The temperature parameter η in Eq. (6) controls how strongly transitions with higher replay scores are prioritized. Smaller values of η give more preference to high-score transitions, while larger values make the sampling distribution closer to Uniform Replay. The weight κ in Eq. (5) controls the relative importance of the reward and cost in the replay score. A larger κ gives more weight to reward, while a smaller κ gives more weight to cost. The performance of the trained agent for all possible choices with $\eta \in \{0.1, 0.5, 1.0, 2.0, 5.0\}$ and $\kappa \in \{0.0, 0.15, 0.35, 0.7, 1.5\}$ has been tested on 4 environments. Among these, Figure 5 demonstrates evaluations with $\kappa = 1.5$, $\forall\eta$ on the SafeHopperVelocity and SafeWalker2dVelocity environments, and with $\eta = 1$, $\forall\kappa$ on the SafeAntVelocity and SafeCarButton1 environments, using SAC-Lagrangian with seven random seeds. The exhibited (and other) results show that $\kappa = 1.5$ and $\eta = 1$ provide a good balance between high reward and safety satisfaction.

6 Conclusion

In this work, we treated the off-policy replay distribution as a control variable for safety. We derive a replay sampling distribution that minimises the variance of cost and reward gradient estimates in off-policy learning. We further proved that this variance-optimal sampler works with any gradient-based safe optimizer, keeps its constrained solution unchanged, and lowers its per-step chance of violating the constraint. The comparative study in our experiments also confirms this across ten different safety gymnasium environments. In the future, we plan to learn the temperature and score weighting online rather than fix them in advance.

7 Declaration of AI-Usage

The authors acknowledge that Large Language Model (LLM) tools were used solely to improve the paper's grammatical correctness. No research ideas, technical contributions, or simulation results were generated using AI.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pp. 22–31. Pmlr, 2017.

Roya Khalili Amirabadi, Mohsen Jalaeian Farimani, and Omid Solaymani Fard. Experience replay in reinforcement learning: A systematic review. *Neurocomputing*, pp. 134403, 2026.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 02 2013. ISBN 9780199535255. doi: 10.1093/acprof:oso/9780199535255.001.0001. URL https://doi.org/10.1093/acprof:oso/9780199535255.001.0001.

Rodrigo Carrasco-Davis, Sebastian Lee, Claudia Clopath, and Will Dabney. Uncertainty prioritized experience replay. *arXiv preprint arXiv:2506.09270*, 2025.

Richard Cheng, Gábor Orosz, Richard M Murray, and Joel W Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 3387–3395, 2019.

Juntao Dai, Jiaming Ji, Long Yang, Qian Zheng, and Gang Pan. Augmented proximal policy optimization for safe reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 7288–7295, 2023.

Ke Fan, Jinpeng Zhang, Xuefeng Zhang, Yunze Wu, Jingyu Cao, Yuan Zhou, and Jianzhu Ma. Safety-polarized and prioritized reinforcement learning. In *Forty-second International Conference on Machine Learning*, 2025.

William Fedus, Prajit Ramachandran, Rishabh Agarwal, Yoshua Bengio, Hugo Larochelle, Mark Rowland, and Will Dabney. Revisiting fundamentals of experience replay. In *International conference on machine learning*, pp. 3061–3071. PMLR, 2020.

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

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. Pmlr, 2018.

Assaf Hallak and Shie Mannor. Consistent on-line off-policy evaluation. In *International Conference on Machine Learning (ICML)*, pp. 1372–1383. PMLR, 2017.

Hossein Hassani, Soodeh Nikan, and Abdallah Shami. Improved exploration–exploitation trade-off through adaptive prioritized experience replay. *Neurocomputing*, 614:128836, 2025.

Somnath Hazra, Pallab Dasgupta, and Soumyajit Dey. Incentivizing safer actions in policy optimization for constrained reinforcement learning. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI '25, 2025. ISBN 978-1-956792-06-5. doi: 10.24963/ijcai.2025/592. URL https://doi.org/10.24963/ijcai.2025/592.

Homayoun Honari, Amir M Soufi Enayati, Mehran Ghafarian Tamizi, and Homayoun Najjaran. Meta sac-lag: Towards deployable safe reinforcement learning via metagradient-based hyperparameter tuning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 619–626. IEEE, 2024.

Jiaming Ji et al. OmniSafe: An infrastructure for accelerating safe reinforcement learning research. *arXiv preprint arXiv:2305.09304*, 2023.

Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*, pp. 2525–2534. PMLR, 2018.

Qiang Liu, Lihong Li, Ziyang Tang, and Dengyong Zhou. Breaking the curse of horizon: Infinite-horizon off-policy estimation. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.

Xu-Hui Liu, Zhenghai Xue, Jingcheng Pang, Shengyi Jiang, Feng Xu, and Yang Yu. Regret minimization experience replay in off-policy reinforcement learning. *Advances in neural information processing systems*, 34:17604–17615, 2021.

Yao Liu, Adith Swaminathan, Alekh Agarwal, and Emma Brunskill. Off-policy policy gradient with state distribution correction. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2019.

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

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc G Bellemare. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. DualDICE: Behavior-agnostic estimation of discounted stationary distribution corrections. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

Yangchen Pan, Jincheng Mei, Amir-massoud Farahmand, Martha White, Hengshuai Yao, Mohsen Rohani, and Jun Luo. Understanding and mitigating the limitations of prioritized experience replay. In *Uncertainty in Artificial Intelligence*, pp. 1561–1571. PMLR, 2022.

Riccardo Poiani, Alberto Maria Metelli, and Marcello Restelli. Truncating trajectories in monte carlo reinforcement learning. In *International Conference on Machine Learning*, pp. 27994–28042. PMLR, 2023.

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

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *International conference on machine learning*, pp. 8583–8592. PMLR, 2020.

Christian Robert Shelton. Importance sampling for reinforcement learning with multiple objectives. 2001.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by pid lagrangian methods. In *International conference on machine learning*, pp. 9133–9143. PMLR, 2020.

Shivakanth Sujit, Somjit Nath, Pedro Braga, and Samira Ebrahimi Kahou. Prioritizing samples in reinforcement learning with reducible loss. *Advances in Neural Information Processing Systems*, 36:23237–23258, 2023.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2000.

Wenwen Tan and Detian Huang. Mp-td3: Multi-pool prioritized experience replay-based asynchronous twin delayed deep deterministic policy gradient algorithm. *IEEE Access*, 12:105268–105280, 2024.

Ren Wang, Kevin Frans, Pieter Abbeel, Sergey Levine, and Alexei Efros. Prioritized generative replay. In *International Conference on Learning Representations*, volume 2025, pp. 47151–47169, 2025.

Tengyu Xu, Yingbin Liang, and Guanghui Lan. CRPO: A new approach for safe reinforcement learning with convergence guarantee. In *International Conference on Machine Learning (ICML)*, 2021.

Qisong Yang, Thiago D Simão, Simon H Tindemans, and Matthijs TJ Spaan. Wcsac: Worst-case soft actor critic for safety-constrained reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10639–10646, 2021.

Yihang Yao, Zuxin Liu, Zhepeng Cen, Jiacheng Zhu, Wenhao Yu, Tingnan Zhang, and Ding Zhao. Constraint-conditioned policy optimization for versatile safe reinforcement learning. *Advances in Neural Information Processing Systems*, 36:12555–12568, 2023.

Haonan Yu, Wei Xu, and Haichao Zhang. Towards safe reinforcement learning with a safety editor policy. *Advances in Neural Information Processing Systems*, 35:2608–2621, 2022.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Ruiyi Zhang, Bo Dai, Lihong Li, and Dale Schuurmans. GenDICE: Generalized offline estimation of stationary values. In *International Conference on Learning Representations (ICLR)*, 2020.

A SUPPLEMENTARY MATERIAL

This supplement gives the proofs of all the theorems, lemmas, propositions, and corollaries used in the main paper. It also provides the implementation details and the hyperparameters used. Lemma 1 shows that the importance weight makes the reward and cost gradient estimates (expectation values) independent of the sampler. Proposition 1 shows that the two scalar tests discussed in Section 4.2 have fixed means that do not depend on the sampler. Theorem 1 establishes an upper bound of the probability of selecting the wrong region for the minibatch estimates of cost and gradients using Cantelli's formula (Boucheron et al., 2013). It proves how this upper bound is directly controlled by the minibatch variance of the test variables from Proposition 1. It then derives the sampler that minimizes these variances for gradient-based policy optimizers. Proposition 2 gives a smoother version of this sampler that maintains replay diversity by adding a trust region. Theorem 2 provides an upper bound on the deviation of the minibatch-based policy gradient update from the ideal update. Corollary 1 shows that the variance-optimal sampler gives the largest guaranteed alignment with the ideal update. Finally, Theorem 3 shows that coupling any safe optimizer with this sampler keeps its constrained solution unchanged and lowers its per-step chance of violating the safety constraint. Throughout the supplementary material, we use the same notation as in the main paper, which is restated below for convenience.

The notation $\langle x,\, y\rangle$ denotes the inner product of the vectors x and y, and $\|x\|$ denotes the norm of x. The notation $\tilde{x}$ denotes the unit vector corresponding to x, given by $\tilde{x} = x/\|x\|$. The quantities $\hat{g}_r$ and $\hat{g}_c$ denote the minibatch estimates of g_r and g_c, respectively. The uniform buffer distribution is denoted by b, and the chosen sampling distribution is denoted by q.

For transition i, the importance weight is obtained by clipping $\beta(s_i)\rho_i$ to the interval $[1/C, C]$ as shown below.

$$w_i = \mathrm{clip}\big(\beta(s_i)\,\rho_i,\ 1/C,\ C\big) = \min\Big\{\,\max\big\{\beta(s_i)\,\rho_i,\ 1/C\big\},\ C\Big\}. \tag{7}$$

Here $C \geq 1$ is a constant, $\rho_i = \pi_\theta(a_i \mid s_i)/\pi_{\mathrm{beh}}(a_i \mid s_i)$ is the ratio of the current policy π_θ to the behavior policy π_{beh}. The term $\beta(s_i) = d^{\pi}(s_i)/d^{\mathrm{beh}}(s_i)$ is the ratio of the current state-visitation distribution to the behavior state-visitation distribution, following Liu et al. (2019) and estimated from buffer samples using the density-ratio method of Liu et al. (2018). The clipping in Eq. 7 ensures that w_i always lies in $[1/C, C]$ by limiting values below $1/C$ to $1/C$ and values above C to C.

Since w_i corrects for both the behavior policy π_{beh} and the behavior state-visitation distribution d^{beh}, it satisfies $\mathbb{E}_{i\sim b}[w_i f] = \mathbb{E}_{(s,a)\sim d^{\pi}}[f]$ for any function f of the transition variables (s_i, a_i). The reward and cost estimates for transition i are $u_i = Q_r(s_i, a_i)\phi_\theta(a_i \mid s_i)$ and $v_i = Q_c(s_i, a_i)\phi_\theta(a_i \mid s_i)$, respectively, so that $g_r = \mathbb{E}_{d^{\pi}}[u]$ and $g_c = \mathbb{E}_{d^{\pi}}[v]$. The corresponding importance-sampled estimators are $X_i = \frac{b(i)}{q(i)} w_i u_i$ and $Y_i = \frac{b(i)}{q(i)} w_i v_i$.

A.1 PROOFS

A.1.1 LEMMA 1: UNBIASED OFF-POLICY REPLAY

We first show that the importance sampling makes the two gradient estimates independent of the sampler, so that replay prioritization can change only their variance.

Lemma 1. *For any sampler q with $q(i) > 0$ wherever the baseline sampler satisfies $b(i) > 0$, $\mathbb{E}_{i\sim q}[X_i] = g_r$ and $\mathbb{E}_{i\sim q}[Y_i] = g_c$. Consequently $\mathbb{E}_{i\sim q}[X_i - \lambda Y_i] = g_r - \lambda g_c$ for every λ, independent of q.*

Proof. By definition, the expectation under the sampler q is the q-weighted sum of X_i over all transitions, as shown below.

$$\mathbb{E}_{i\sim q}[X_i] = \sum_i q(i)\, X_i.$$

We substitute the definition $X_i = \frac{b(i)}{q(i)} w_i u_i$ into this sum, as shown below.

$$\mathbb{E}_{i\sim q}[X_i] = \sum_i q(i)\, \frac{b(i)}{q(i)}\, w_i u_i.$$

In each term the factor $q(i)$ in front cancels the factor $q(i)$ in the denominator. This leaves a sum that no longer contains q, as shown below.

$$\mathbb{E}_{i\sim q}[X_i] = \sum_i b(i)\, w_i u_i.$$

The right-hand side is, by definition, the expectation of $w_i u_i$ when the i-th transition is drawn from the replay buffer with distribution b. This is written below.

$$\sum_i b(i)\, w_i u_i = \mathbb{E}_{i\sim b}[w_i u_i].$$

We now apply the importance-weighting identity. Because $w_i = \beta(s_i)\rho_i$ corrects both the behavior action law and the behavior state visitation, replacing f by u in $\mathbb{E}_{i\sim b}[w_i f] = \mathbb{E}_{(s,a)\sim d^\pi}[f]$ gives the expectation below.

$$\mathbb{E}_{i\sim b}[w_i u_i] = \mathbb{E}_{(s,a)\sim d^\pi}[u] = g_r.$$

The computations above therefore give $\mathbb{E}_{i\sim q}[X_i] = g_r$. The same three steps with v_i in place of u_i give $\mathbb{E}_{i\sim q}[Y_i] = g_c$. Finally, by linearity of expectation we subtract the two results, as shown below.

$$\mathbb{E}_{i\sim q}[X_i - \lambda Y_i] = \mathbb{E}_{i\sim q}[X_i] - \lambda\, \mathbb{E}_{i\sim q}[Y_i] = g_r - \lambda g_c.$$

This holds for every λ and does not depend on q. □

The sampler q cancels before the average is formed, so it can never bias the two gradients. The gradient update $\mathcal{F}(g_r, g_c)$ that any optimizer builds from g_r and g_c, and therefore its zero set, is likewise unchanged by q. With the mean fixed, all that remains for the sampler to influence is the variance, which we study next through two scalar diagnostics.

A.1.2 Proposition 1: Fixed Means of the Scalar Estimators

We write $\tilde{g}_c = g_c/\|g_c\|$ and $\tilde{g}_r^+ = g_r^+/\|g_r^+\|$ with $g_r^+ = g_r - \frac{\omega}{\|g_c\|^2} g_c$ and $\omega = \langle g_r,\, g_c\rangle$. We define the scores $s_i^c = w_i\langle v_i,\, \tilde{g}_c\rangle$ and $s_i^r = w_i\langle u_i,\, \tilde{g}_r^+\rangle$ and the estimators $Z^c = \frac{b(i)}{q(i)} s_i^c$ and $Z^r = \frac{b(i)}{q(i)} s_i^r$.

Proposition 1. $\mathbb{E}_{i\sim q}[Z^c] = \|g_c\|$ *and* $\mathbb{E}_{i\sim q}[Z^r] = \|g_r^+\|$ *for every* q.

Proof. For the cost side we first cancel the ratio $b(i)/q(i)$ exactly as in Lemma 1, which turns the q-expectation into a b-weighted sum of the scores, as shown below.

$$\mathbb{E}_{i\sim q}[Z^c] = \sum_i q(i)\, \frac{b(i)}{q(i)}\, s_i^c = \sum_i b(i)\, s_i^c = \sum_i b(i)\, w_i\langle v_i,\, \tilde{g}_c\rangle.$$

By linearity of the inner product, with $\tilde{g}_c$ fixed, we get the following.

$$\sum_i b(i)\, w_i\langle v_i,\, \tilde{g}_c\rangle = \Big\langle \sum_i b(i) w_i v_i,\, \tilde{g}_c \Big\rangle.$$

By Lemma 1, $\sum_i b(i) w_i v_i = g_c$. Substituting this in $\Big\langle \sum_i b(i) w_i v_i,\, \tilde{g}_c \Big\rangle$ and then using $\tilde{g}_c = g_c/\|g_c\|$ gives the following.

$$\mathbb{E}_{i\sim q}[Z^c] = \langle g_c,\, \tilde{g}_c\rangle = \frac{\langle g_c,\, g_c\rangle}{\|g_c\|} = \frac{\|g_c\|^2}{\|g_c\|} = \|g_c\|.$$

For the reward side the identical cancellation and linearity steps give an inner product with the reward gradient, as shown below.

$$\mathbb{E}_{i\sim q}[Z^r] = \Big\langle \sum_i b(i) w_i u_i,\, \tilde{g}_r^+ \Big\rangle = \langle g_r,\, \tilde{g}_r^+\rangle.$$

To evaluate this we first compute the term $\langle g_r,\, g_r^+\rangle$ using the definition $g_r^+ = g_r - \frac{\omega}{\|g_c\|^2} g_c$, and $\langle g_r,\, g_c\rangle = \omega$ as shown below.

$$\langle g_r,\, g_r^+\rangle = \Big\langle g_r,\, g_r - \tfrac{\omega}{\|g_c\|^2} g_c \Big\rangle = \|g_r\|^2 - \frac{\omega}{\|g_c\|^2}\langle g_r,\, g_c\rangle = \|g_r\|^2 - \frac{\omega^2}{\|g_c\|^2},$$

We next expand the squared norm of g_r^+ from the same definition, as shown below.

$$\|g_r^+\|^2 = \Big\langle g_r - \tfrac{\omega}{\|g_c\|^2} g_c,\, g_r - \tfrac{\omega}{\|g_c\|^2} g_c \Big\rangle = \|g_r\|^2 - 2\frac{\omega}{\|g_c\|^2}\langle g_r,\, g_c\rangle + \frac{\omega^2}{\|g_c\|^4}\|g_c\|^2 = \|g_r\|^2 - \frac{\omega^2}{\|g_c\|^2}.$$

The two equations above have the same right-hand side, so $\langle g_r,\, g_r^+\rangle = \|g_r^+\|^2$. Dividing by $\|g_r^+\|$ and using $\tilde{g}_r^+ = g_r^+/\|g_r^+\|$ gives the reward mean below.

$$\mathbb{E}_{i\sim q}[Z^r] = \langle g_r,\, \tilde{g}_r^+\rangle = \frac{\langle g_r,\, g_r^+\rangle}{\|g_r^+\|} = \frac{\|g_r^+\|^2}{\|g_r^+\|} = \|g_r^+\|$$

□

The two means are fixed constants that do not depend on q. Only the variances $\sigma_c^2(q)$ and $\sigma_r^2(q)$ depend on the sampler.

A.1.3 Theorem 1: The Variance-Optimal Sampler

We compute $\mathbb{E}_q[(Z^c)^2]$ by drawing transition i with probability $q(i)$ and squaring the value $\frac{b(i)}{q(i)} s_i^c$, which gives the following.

$$\mathbb{E}_{i\sim q}[(Z^c)^2] = \sum_i q(i)\Big(\frac{b(i)}{q(i)} s_i^c\Big)^2 = \sum_i q(i)\,\frac{b(i)^2}{q(i)^2}(s_i^c)^2 = \sum_i \frac{b(i)^2}{q(i)}(s_i^c)^2.$$

Similarly $\mathbb{E}_q[(Z^r)^2]$ can also be computed. Subtracting the squared means $\|g_c\|^2$ and $\|g_r^+\|^2$ of Z^c and Z^r respectively obtained from Proposition 1, and subtracting it from $\mathbb{E}_q[(Z^c)^2]$ and $\mathbb{E}_q[(Z^r)^2]$, gives the two variances below.

$$\sigma_c^2(q) = \sum_i \frac{b(i)^2}{q(i)}(s_i^c)^2 - \|g_c\|^2, \qquad \sigma_r^2(q) = \sum_i \frac{b(i)^2}{q(i)}(s_i^r)^2 - \|g_r^+\|^2.$$

Theorem 1. *Let $\mathcal{Q} = \{q \in \Delta^N : q(i) > 0$ whenever $b(i)[(s_i^c)^2 + \kappa(s_i^r)^2] > 0\}$. The objective $\sigma_c^2(q) + \kappa\,\sigma_r^2(q)$ is minimized over $\mathcal{Q}$ uniquely by $q^\star(i) = \sqrt{A_i}/\sum_j \sqrt{A_j}$ with $A_i = b(i)^2[(s_i^c)^2 + \kappa(s_i^r)^2]$, and its minimum value is $\big(\sum_i \sqrt{A_i}\big)^2 - \|g_c\|^2 - \kappa\|g_r^+\|^2$.*

Proof. We first add the two variance formulas as shown below.

$$\sigma_c^2(q) + \kappa\sigma_r^2(q) = \Phi(q) - \big(\|g_c\|^2 + \kappa\|g_r^+\|^2\big), \qquad \Phi(q) = \sum_i \frac{A_i}{q(i)}.$$

Here, $A_i = b(i)^2[(s_i^c)^2 + \kappa(s_i^r)^2]$. The term $\big(\|g_c\|^2 + \kappa\|g_r^+\|^2\big)$ is constant, as shown in Proposition 1. Therefore, minimizing the objective is equivalent to minimizing Φ. We now try to find the lower-bound of Φ. For this we apply the Cauchy–Schwarz inequality to the sequences $x_i = \sqrt{A_i/q(i)}$ and $y_i = \sqrt{q(i)}$, for which the product is $x_i y_i = \sqrt{A_i}$, as shown below.

$$\Big(\sum_i \sqrt{A_i}\Big)^2 = \Big(\sum_i x_i y_i\Big)^2 \le \Big(\sum_i x_i^2\Big)\Big(\sum_i y_i^2\Big).$$

We now evaluate the two factors on the right. Using $x_i^2 = A_i/q(i)$ and $y_i^2 = q(i)$ together with the fact that $\sum_i q(i) = 1$ on the above expression gives the value below.

$$\Big(\sum_i x_i^2\Big)\Big(\sum_i y_i^2\Big) = \Big(\sum_i \frac{A_i}{q(i)}\Big)\Big(\sum_i q(i)\Big) = \Phi(q)\cdot 1 = \Phi(q).$$

Combining the last two equations gives $\Phi(q) \ge \big(\sum_i \sqrt{A_i}\big)^2$, a lower bound that does not depend on q. Equality in Cauchy–Schwarz holds exactly when x_i is proportional to y_i. Writing $x_i = \lambda y_i$ gives $\sqrt{A_i/q(i)} = \lambda\sqrt{q(i)}$, hence $A_i/q(i) = \lambda^2 q(i)$, hence $q(i)^2 \propto A_i$, which means $q(i) \propto \sqrt{A_i}$.

After normalization we get: $q^\star(i) = \sqrt{A_i}/\sum_j \sqrt{A_j}$. To obtain the minimum value of $\Phi(q)$ we substitute $q^\star$ into the expression of $\Phi(q)$, where $S = \sum_j \sqrt{A_j}$ as shown below.

$$\Phi(q^\star) = \sum_i \frac{A_i}{q^\star(i)} = S \sum_i \sqrt{A_i} = S \cdot S = \Big(\sum_i \sqrt{A_i} \Big)^2.$$

Subtracting the constant $\|g_c\|^2 + \kappa \|g_r^+\|^2$ gives the stated minimum value of $\big(\sum_i \sqrt{A_i} \big)^2 - \|g_c\|^2 - \kappa \|g_r^+\|^2$. Each term $A_i/q(i)$ with $A_i > 0$ is strictly convex in $q(i)$, so Φ is a sum of strictly convex functions and is therefore strictly convex, which makes the minimizer unique. □

A.1.4 Proposition 2: Trust-Region based Sampler

The exact minimizer $q^\star(i) \propto b(i)\rho_i$ with $\rho_i = \sqrt{(s_i^c)^2 + \kappa (s_i^r)^2}$ can put almost all of its probability on a few transitions, which hurts replay diversity. Adding the KL trust region directly to the variance objective restores diversity but gives an equation with no closed-form solution. To tackle this, instead we keep the same goal of favouring high-score transitions but apply the trust region constraint to a simpler linear objective, as shown below, preserving both the closed-form solution and sample diversity.

$$\max_{q \in \Delta^N} \sum_i q(i)\rho_i \quad \text{s.t.} \quad D_{\mathrm{KL}}(q\|b) \le \epsilon. \tag{8}$$

Proposition 2. *The solution of Eq.* (8) *is* $q_\epsilon^\star(i) = b(i)e^{\rho_i/\eta}/\sum_j b(j)e^{\rho_j/\eta}$*, where* $\eta > 0$ *is the multiplier chosen so that the constraint binds.*

Proof. We introduce a multiplier $\eta \ge 0$ for the trust region and a multiplier ν for the normalization constraint $\sum_i q(i) = 1$. The Lagrangian is given below.

$$\mathcal{L} = \sum_i q(i)\rho_i - \eta \Big(\sum_i q(i) \log \tfrac{q(i)}{b(i)} - \epsilon \Big) - \nu \Big(\sum_i q(i) - 1 \Big).$$

We differentiate with respect to a single $q(i)$ and obtain the equation below.

$$\frac{\partial \mathcal{L}}{\partial q(i)} = \rho_i - \eta \Big(\log \tfrac{q(i)}{b(i)} + 1 \Big) - \nu = 0.$$

Solving for the log-ratio gives the expression below.

$$\log \tfrac{q(i)}{b(i)} = \frac{\rho_i - \nu}{\eta} - 1 = \frac{\rho_i}{\eta} - \frac{\eta + \nu}{\eta}.$$

Exponentiating both sides gives $q(i) = b(i)\, e^{\rho_i/\eta}\, e^{-(\eta+\nu)/\eta}$. The factor $e^{-(\eta+\nu)/\eta}$ does not depend on i, so we write it as a constant c fixed by $\sum_i q(i) = 1$, which yields the stated form below.

$$q_\epsilon^\star(i) = \frac{b(i)e^{\rho_i/\eta}}{\sum_j b(j)e^{\rho_j/\eta}}.$$

The objective is linear and the KL term is strictly convex, so the stationary point is the unique maximizer, and η is set by requiring $D_{\mathrm{KL}}(q_\epsilon^\star\|b) = \epsilon$. □

As $\eta \to \infty$ every $e^{\rho_i/\eta} \to 1$ and $q_\epsilon^\star \to b$, which is full diversity, while as $\eta \to 0^+$ the term with the largest ρ_i dominates and $q_\epsilon^\star$ concentrates on $\arg\max_i \rho_i$, which is greedy prioritization. Since $e^{(\cdot)} > 0$, every transition with $b(i) > 0$ keeps positive probability, so no sample is ever dropped. The proposed sampler therefore lowers the safety variance while retaining diversity.

A.1.5 Assumptions on the Base Optimizer

We now specify the assumptions on the update rule $\mathcal{F}$ needed to show how replay sampling affects the policy update. This class includes Lagrangian, primal-dual, and CRPO methods.

- (F1) *Homogeneity.* $\mathcal{F}(tg_r, tg_c) = t\,\mathcal{F}(g_r, g_c)$ for every $t > 0$.
- (F2) *Piecewise-linear branches.* The switching surfaces divide gradient space into finitely many regions $R_1, \dots, R_m$, and on region R_k the map is linear, $\mathcal{F}(g_r, g_c) = A_k g_r + B_k g_c$. We write $L_{\mathcal{F}} = \max_k(\|A_k\| + \|B_k\|)$.
- (F3) *Bounded estimates.* $\|\hat{g}_r\|, \|\hat{g}_c\| \le \bar{G}$ almost surely, which the clipped weights $w_i \in [1/C, C]$ guarantee.

A minibatch of size n under the distribution q gives: $\hat{g}_r = \frac{1}{n}\sum X_i$ and $\hat{g}_c = \frac{1}{n}\sum Y_i$. Let $R^\star$ be the region that contains the true pair (g_r, g_c), so that $\Delta^\star = \mathcal{F}(g_r, g_c) = A^\star g_r + B^\star g_c$. The realized step is $\Delta_q = \mathcal{F}(\hat{g}_r, \hat{g}_c)$ and the effective gradient is $\bar{\Delta}(q) = \mathbb{E}_q[\Delta_q]$. The flip event is $\mathcal{E} = \{(\hat{g}_r, \hat{g}_c) \notin R^\star\}$ and $P_{\text{flip}}(q) = \Pr_q(\mathcal{E})$. A flip is detected by the two sign tests of Proposition 1. Each test simply reads the sign of a variable averaged over a minibatch. The cost-descent test reads the sign of Z^c, whose mean is $\|g_c\|$ and variance is σ_c^2, and the reward-progress test reads the sign of Z^r, whose mean is $\|g_r^+\|$ and variance is σ_r^2. Both means are positive, so the correct outcome of each test is a positive sign. A flip is the event that a single minibatch turns a sign negative, that is $Z^c \le 0$ or $Z^r \le 0$, meaning the estimate points the wrong way. The branch-decision variances are therefore exactly the σ_c^2 and σ_r^2 that the sampler minimizes.

A.1.6 Theorem 2: Effective-Gradient Dominance

Theorem 2. *Under assumptions (F1) to (F3), the following two bounds hold.*

$$\left\|\bar{\Delta}(q) - \Delta^\star\right\| \;\le\; 2L_{\mathcal{F}}\,\bar{G}\,P_{\text{flip}}(q), \tag{9}$$

$$P_{\text{flip}}(q) \;\le\; \frac{\sigma_c^2(q)}{\sigma_c^2(q) + \|g_c\|^2} + \frac{\sigma_r^2(q)}{\sigma_r^2(q) + \|g_r^+\|^2}. \tag{10}$$

Both ratios are increasing in σ_c^2 and σ_r^2, so the variance-optimal sampler $q^\star$ of Theorem 1 minimizes the right-hand sides over all admissible samplers.

Proof. Let $\mathcal{E}$ denote the event that a minibatch flips. Its indicator $\mathbf{1}_{\mathcal{E}}$ equals 1 if the minibatch flips and 0 otherwise. Similarly, $\mathbf{1}_{\mathcal{E}^c}$ equals 1 if the minibatch does not flip and 0 otherwise. Every minibatch falls into exactly one of these cases, so $\mathbf{1} = \mathbf{1}_{\mathcal{E}^c} + \mathbf{1}_{\mathcal{E}}$, and we use this identity to split the average error into the no-flip case and the flip case. The deviation of $\bar{\Delta}(q)$ from $\Delta^\star$ is shown below.

$$\bar{\Delta}(q) - \Delta^\star = \underbrace{\mathbb{E}_q\big[(\mathcal{F}(\hat{g}_r, \hat{g}_c) - \Delta^\star)\mathbf{1}_{\mathcal{E}^c}\big]}_{T_1} + \underbrace{\mathbb{E}_q\big[(\mathcal{F}(\hat{g}_r, \hat{g}_c) - \Delta^\star)\mathbf{1}_{\mathcal{E}}\big]}_{T_2}.$$

Consider T_1 first. On $\mathcal{E}^c$ the estimate lies in the correct region $R^\star$, so $\mathcal{F}(\hat{g}_r, \hat{g}_c) = A^\star\hat{g}_r + B^\star\hat{g}_c$. Using $\mathbb{E}_q[\hat{g}_r] = g_r$ and $\mathbb{E}_q[\hat{g}_c] = g_c$ from Lemma 1 together with $\Delta^\star = A^\star g_r + B^\star g_c$, the expectation of this term is zero, as shown below.

$$\mathbb{E}_q\big[A^\star\hat{g}_r + B^\star\hat{g}_c - \Delta^\star\big] = A^\star g_r + B^\star g_c - \Delta^\star = \Delta^\star - \Delta^\star = 0.$$

We now write $\mathbf{1}_{\mathcal{E}^c} = 1 - \mathbf{1}_{\mathcal{E}}$ in T_1 and use this zero-average fact, as shown below.

$$\begin{aligned} T_1 &= \mathbb{E}_q\big[(A^\star\hat{g}_r + B^\star\hat{g}_c - \Delta^\star)(1 - \mathbf{1}_{\mathcal{E}})\big] \\ &= \underbrace{\mathbb{E}_q\big[A^\star\hat{g}_r + B^\star\hat{g}_c - \Delta^\star\big]}_{=0} - \mathbb{E}_q\big[(A^\star\hat{g}_r + B^\star\hat{g}_c - \Delta^\star)\mathbf{1}_{\mathcal{E}}\big] \\ &= -\,\mathbb{E}_q\big[(A^\star\hat{g}_r + B^\star\hat{g}_c - \Delta^\star)\mathbf{1}_{\mathcal{E}}\big]. \end{aligned}$$

Now consider T_2. On $\mathcal{E}$ the estimate lies in some wrong region $\hat{k}$, so $\mathcal{F}(\hat{g}_r, \hat{g}_c) = A_{\hat{k}}\hat{g}_r + B_{\hat{k}}\hat{g}_c$, which gives T_2 below.

$$T_2 = \mathbb{E}_q\big[(A_{\hat{k}}\hat{g}_r + B_{\hat{k}}\hat{g}_c - \Delta^\star)\mathbf{1}_{\mathcal{E}}\big].$$

Adding T_1 and T_2, both are expectations over the same event $\mathcal{E}$, so we combine them into one and the two $-\Delta^\star$ terms cancel inside the bracket, as shown below.

$$\begin{aligned}\bar{\Delta}(q) - \Delta^\star &= \mathbb{E}_q\big[(A_{\hat{k}}\hat{g}_r + B_{\hat{k}}\hat{g}_c - \Delta^\star)\mathbf{1}_\mathcal{E}\big] - \mathbb{E}_q\big[(A^\star\hat{g}_r + B^\star\hat{g}_c - \Delta^\star)\mathbf{1}_\mathcal{E}\big] \\ &= \mathbb{E}_q\Big[\big[(A_{\hat{k}}\hat{g}_r + B_{\hat{k}}\hat{g}_c - \Delta^\star) - (A^\star\hat{g}_r + B^\star\hat{g}_c - \Delta^\star)\big]\mathbf{1}_\mathcal{E}\Big] \\ &= \mathbb{E}_q\big[\big((A_{\hat{k}} - A^\star)\hat{g}_r + (B_{\hat{k}} - B^\star)\hat{g}_c\big)\mathbf{1}_\mathcal{E}\big].\end{aligned}$$

We now bound the size of the term inside this expectation on the flip event. Applying the triangle inequality and then $\|\hat{g}_r\|, \|\hat{g}_c\| \le \bar{G}$ from assumption (F3) gives the bound below.

$$\big\|(A_{\hat{k}} - A^\star)\hat{g}_r + (B_{\hat{k}} - B^\star)\hat{g}_c\big\| \le (\|A_{\hat{k}}\| + \|A^\star\|)\bar{G} + (\|B_{\hat{k}}\| + \|B^\star\|)\bar{G}.$$

By assumption (F2) each region satisfies $\|A_{\hat{k}}\| + \|B_{\hat{k}}\| \le L_\mathcal{F}$ and $\|A^\star\| + \|B^\star\| \le L_\mathcal{F}$, so the right-hand side is at most $2L_\mathcal{F}\bar{G}$. Since $\|\mathbb{E}[\cdot]\| \le \mathbb{E}[\|\cdot\|]$, and the quantity inside the expectation is nonzero only when a flip occurs (through the factor $\mathbf{1}_\mathcal{E}$), we obtain (9) as shown below.

$$\|\bar{\Delta}(q) - \Delta^\star\| \le 2L_\mathcal{F}\bar{G}\,\mathbb{E}_q[\mathbf{1}_\mathcal{E}] = 2L_\mathcal{F}\bar{G}\,P_{\text{flip}}(q).$$

A flip event happens only if at least one of the two tests comes out with the wrong sign, that is $Z^c \le 0$ or $Z^r \le 0$. Since the probability of either event is at most the sum of their probabilities, we obtain the inequality below.

$$P_{\text{flip}}(q) \le P(Z^c \le 0) + P(Z^r \le 0).$$

Each statistic has a positive mean, so the unsafe cost event is $\{Z^c \le 0\} = \{Z^c \le \|g_c\| - \|g_c\|\}$. Applying Cantelli's inequality (Poiani et al., 2023) with $t = \|g_c\|$, and the same step for the reward statistic with $t = \|g_r^+\|$, gives the two bounds below.

$$P(Z^c \le 0) \le \frac{\sigma_c^2}{\sigma_c^2 + \|g_c\|^2}, \qquad P(Z^r \le 0) \le \frac{\sigma_r^2}{\sigma_r^2 + \|g_r^+\|^2}.$$

Adding these two bounds gives Eq. (10). Finally, each ratio $\sigma^2/(\sigma^2 + m^2)$ has derivative $m^2/(\sigma^2 + m^2)^2 > 0$ in σ^2, so it grows with the variance, and Theorem 1 makes $q^\star$ the unique minimizer of σ_c^2 and of $\sigma_c^2 + \kappa\sigma_r^2$, hence of the bound. □

A.1.7 Corollary 1: Larger Effective Gradient

Corollary 1. *Let $a = q^\star$ and let b be any sampler with $\sigma_c^2(a) \le \sigma_c^2(b)$ and $\sigma_r^2(a) \le \sigma_r^2(b)$. Then $\langle\Delta^\star, \bar{\Delta}(a)\rangle \ge \|\Delta^\star\|^2 - 2L_\mathcal{F}\bar{G}\,\|\Delta^\star\|\,P_{\text{flip}}(b)$.*

Proof. Write $\bar{\Delta}(q) = \Delta^\star - (\Delta^\star - \bar{\Delta}(q))$ and expand the inner product by linearity, as shown below.

$$\begin{aligned}\langle\Delta^\star, \bar{\Delta}(q)\rangle &= \langle\Delta^\star, \Delta^\star - (\Delta^\star - \bar{\Delta}(q))\rangle \\ &= \langle\Delta^\star, \Delta^\star\rangle - \langle\Delta^\star, \Delta^\star - \bar{\Delta}(q)\rangle \\ &= \|\Delta^\star\|^2 - \langle\Delta^\star, \Delta^\star - \bar{\Delta}(q)\rangle.\end{aligned}$$

We now bound the subtracted term $\langle\Delta^\star, \Delta^\star - \bar{\Delta}(q)\rangle$. We apply the Cauchy–Schwarz inequality to the inner product, use the fact that $\|\Delta^\star - \bar{\Delta}(q)\| = \|\bar{\Delta}(q) - \Delta^\star\|$, and finally use Eq. (9) to obtain the following.

$$\begin{aligned}\langle\Delta^\star, \Delta^\star - \bar{\Delta}(q)\rangle &\le \|\Delta^\star\|\,\|\Delta^\star - \bar{\Delta}(q)\| \\ &= \|\Delta^\star\|\,\|\bar{\Delta}(q) - \Delta^\star\| \\ &\le \|\Delta^\star\| \cdot 2L_\mathcal{F}\bar{G}\,P_{\text{flip}}(q).\end{aligned}$$

Substituting this upper bound into the identity above gives the lower bound below.

$$\langle\Delta^\star, \bar{\Delta}(q)\rangle \ge \|\Delta^\star\|^2 - 2L_\mathcal{F}\bar{G}\,\|\Delta^\star\|\,P_{\text{flip}}(q).$$

This lower bound is decreasing in $P_{\text{flip}}(q)$. Applying it to a and to b, and using $P_{\text{flip}}(a) \le P_{\text{flip}}(b)$ from Theorem 2, gives the chain below.

$$\langle\Delta^\star, \bar{\Delta}(a)\rangle \ge \|\Delta^\star\|^2 - 2L_\mathcal{F}\bar{G}\,\|\Delta^\star\|\,P_{\text{flip}}(a) \ge \|\Delta^\star\|^2 - 2L_\mathcal{F}\bar{G}\,\|\Delta^\star\|\,P_{\text{flip}}(b).$$

Reading off the two ends of this chain gives $\langle\Delta^\star, \bar{\Delta}(a)\rangle \ge \|\Delta^\star\|^2 - 2L_\mathcal{F}\bar{G}\,\|\Delta^\star\|\,P_{\text{flip}}(b)$, which is the claim. □

Among all samplers, $q^\star$ gives the largest guaranteed alignment with the ideal step $\Delta^\star$.

A.1.8 THEOREM 3: THE SAFE REINFORCEMENT LEARNING ALGORITHM

Theorem 3. *Let $\mathcal{F}$ be a safe gradient optimizer that converges to a KKT point of the CMDP when run with exact gradients. Wrapping $\mathcal{F}$ with the unbiased variance-optimal sampler as implemented in Algorithm 1, produces an algorithm with the same constrained stationary set as $\mathcal{F}$.*

Proof. Same stationary set. By Lemma 1, the means of the gradients g_r and g_c do not depend on the sampler q. Therefore, the optimizer $\mathcal{F}$ receives the same average gradients as with other buffer sampling methods, such as uniform sampling and TD-error prioritized sampling. Thus, its update rule, fixed points, and KKT conditions, including feasibility and complementary slackness, remain unchanged. The proposed sampler cannot add or remove a stationary point, so the algorithm 1 has exactly the constrained stationary set of $\mathcal{F}$.

Convergence and per-step safety. The updates $\theta_{k+1} = \theta_k + \alpha\mathcal{F}(\hat{g_r}, \hat{g_c})$ are the gradient-based optimizer's own updates using unbiased gradient estimates. Therefore, the convergence guarantee of these optimizers to the KKT points still holds. Our proposed sampler affects only the noise in these estimates, measured by σ_c^2 and σ_r^2, and Theorem 1 shows that it minimizes these variances. □

Safety is thus not re-derived. It is the gradient-based optimizer's own guarantee, shown undisturbed by the sampler and made more accurate at each step. Because the correction $w_i = \beta(s_i)\rho_i$ keeps the fed gradients unbiased for the current policy, the constrained stationary set is preserved even though the buffer mixes states and actions from many past policies.

A.2 IMPLEMENTATION AND EXPERIMENTAL PROTOCOLS

This section records all the settings used to produce the reported results so that the experiments can be reproduced exactly. The complete set of files required to reproduce the results is available at: `https://anonymous.4open.science/r/variance-optimal-replay-0606/README.md`.

Base learner. VOSR is a replay module rather than a policy rule. It maintains separate reward and cost critics and hands the importance sampled, variance-optimal gradient estimates $\hat{g_r}, \hat{g_c}$ to an arbitrary base safe optimizer $\mathcal{F}$, namely SAC-Lagrangian, CRPO, or PCRPO in our experiments. It carries no Lagrange multiplier or slack of its own, since those belong to whichever $\mathcal{F}$ is chosen. The fraction $\frac{b(i)}{q_\epsilon^\star(i)} w_i$ multiplies every critic and actor gradient contribution, which keeps Lemma 1 in force. The reward critic is a standard twin network, and the cost critic is a three-member ensemble whose spread across heads also serves as the priority signal for the Uncertainty-PER baseline.

State-visitation ratio estimation. The weight $w_i = \beta(s_i)\rho_i$ combines the action ratio $\rho_i = \pi_\theta(a_i \mid s_i)/\pi_{\text{beh}}(a_i \mid s_i)$, computed directly from the stored behavior log-probability, with the state-visitation ratio $\beta(s_i) = d^\pi(s_i)/d^{\text{beh}}(s_i)$. We estimate β with the infinite-horizon density-ratio objective of Liu et al. (2019; 2018): a small two-hidden-layer network $\beta_\psi : \mathcal{S} \to \mathbb{R}_{>0}$ (128 units, softplus output) is fit by minimizing the kernelized loss $\max_f L(\beta_\psi, f)$ restricted to a unit ball in an RKHS, which admits a closed-form inner maximization. The estimator is refreshed every 250 gradient steps on a reference batch of 256 transitions using Adam with learning rate 3×10^{-4}, and the combined weight $w_i = \beta_\psi(s_i)\rho_i$ is clipped to $[1/C, C] = [0.1, 10]$ before importance sampling. Any behavior-agnostic stationary-distribution estimator, e.g. DualDICE (Nachum et al., 2019) or GenDICE (Zhang et al., 2020), can be substituted for β_ψ without altering the sampler or any of the theoretical guarantees.

Shared agent and network settings. The values in Table 2 are shared across all methods and are not overridden per method. The policy is a Gaussian network with two hidden layers of 128 units and $\tanh$ activations, and the two critics use two hidden layers of 128 units with ReLU activations. The log standard deviation of the policy is clamped to $[-2, 2]$. The per-step cost exponential moving average (EMA) uses coefficient 0.99, and the episodic cost EMA used by VOSR uses coefficient 0.9. Each cost-critic ensemble member is included in the critic loss with probability 0.85 per sample, which acts as a lightweight ensemble dropout.

Table 2: Shared agent and network hyperparameters, applied to every method.

Hyperparameter	Value
Discount γ	0.99
Target smoothing τ (Polyak)	0.005
Actor learning rate	3×10^{-4}
Critic learning rate (both critics)	3×10^{-4}
Replay buffer capacity	300,000
Importance-weight clip bound C	10.0 (weights in $[0.1, 10]$)
Reward critic	twin network ($k = 2$)
Cost critic ensemble size	3
Candidate minibatch pool	512
Reference batch (VOSR)	256
Training minibatch size	256
Warmup (uniform-random) steps	1,000
Train frequency	every 4 environment steps
Policy hidden layers	2×128, $\tanh$
Critic hidden layers	2×128, ReLU
Policy $\log \sigma$ range	$[-2.0,\ 2.0]$
State-ratio network β_ψ	2×128, softplus
State-ratio learning rate	3×10^{-4}
State-ratio refresh interval	every 250 steps

Environments and training budgets. We evaluate on eleven Safety Gymnasium environments, six velocity locomotion tasks and five navigation tasks, under a uniform cost budget of 25. Table 3 lists the total training steps, evaluation interval, and maximum episode length used for each environment. Every environment and method is run with seven seeds $\{0, 1, \ldots, 6\}$, each evaluation checkpoint uses three evaluation episodes, and learning and cost curves report the mean and one standard deviation over the seven seeds. The full comparison covers three base optimizers and five samplers per optimizer, which gives 15 methods per environment. The main paper displays representative showcase environments for each optimizer; the remaining environments follow the same protocol.

Table 3: Per-environment training budgets. The cost budget is $d = 25$ for all environments.

Environment	Total steps	Eval interval	Max ep. length
SafeAntVelocity	32,000	2,000	1000
SafeHalfCheetahVelocity	32,000	2,000	1000
SafeHopperVelocity	32,000	2,000	1000
SafeHumanoidVelocity	32,000	2,000	1000
SafeSwimmerVelocity	32,000	2,000	1000
SafeWalker2dVelocity	32,000	2,000	1000
SafePointButton1	20,000	1,000	1000
SafePointPush1	20,000	1,000	1000
SafePointGoal2	20,000	1,000	1000
SafeCarButton1	20,000	1,000	1000
SafeCarGoal1	20,000	1,000	1000

Base-optimizer settings. Each base optimizer keeps its own published rule, and VOSR does not alter its internal logic. SAC-Lagrangian updates its multiplier as $\lambda \leftarrow \max(0, \lambda + \eta_\lambda(\bar{c} - d_{\text{eff}}))$, where $\bar{c}$ is the tracked cost and d_{eff} is the effective budget. CRPO follows the reward gradient while the tracked cost is within the effective budget and switches to $-g_c$ otherwise. PCRPO follows the reward gradient while safe, and otherwise projects the reward gradient orthogonal to the cost gradient and subtracts a weighted cost gradient. Table 4 lists their settings. The effective budget is $d_{\text{eff}} = d \cdot \text{margin}$, and the per-step budget is $d_{\text{step}} = d/(\text{max episode length})$.

VOSR sampler and per-environment tiers. The tempered sampler of Proposition 2 uses temperature $\eta = 1.0$ for every run in the main campaign. The priority score is $\rho_i =$

Table 4: Base-optimizer settings. Baseline denotes non-VOSR runs and VOSR denotes VOSR-wrapped runs.

Optimizer	Parameter	Value
SAC-Lagrangian	multiplier rate η_λ	0.01 (baseline), 10^{-4} (VOSR)
	initial multiplier λ_0	0.0
	multiplier cap $\lambda_{\max}$	none (baseline), 30.0 (VOSR)
CRPO	branch-switch tolerance	0.0
PCRPO	cost-gradient weight	1.0
All	safety margin (baseline)	1.0

$\sqrt{(s_i^c)^2 + \kappa(s_i^r)^2 + 10^{-12}}$, the cost and reward scores s_i^c, s_i^r are robustly clipped with a median-absolute-deviation multiplier of 6.0, and the tempered softmax standardizes ρ_i to zero mean and unit standard deviation within the candidate pool before dividing by η, so that η acts as a scale-free knob measured in standard deviations of score difference. The reward-variance weight κ and the optimizer margin are set per environment tier, as listed in Table 5. Non-VOSR baselines always use margin 1.0, and their three prioritized variants (TD-PER, Safety-PER, Uncertainty-PER) raise each priority to the power 0.6 after adding a floor of 10^{-3}, following standard prioritized replay.

Table 5: Per-environment VOSR tiers. Each tier sets the optimizer margin and the reward-variance weight κ. Baselines use margin 1.0 and do not read κ.

Tier	Margin	κ	Environments
Severe	0.15	0.15	PointButton1, PointPush1, PointGoal2, CarButton1, CarGoal1, Swimmer
Mild	0.50	0.35	Ant, HalfCheetah
Reward-max	0.80	0.70	Hopper, Humanoid, Walker2d

Ablation protocol. The ablations in the main paper isolate the temperature η and the reward-variance weight κ. Every ablation uses SAC-Lagrangian wrapped with VOSR, seven seeds $\{0, \ldots, 6\}$, and the four environments SafeAntVelocity, SafeCarButton1, SafeHopperVelocity, and SafeWalker2dVelocity, with all other settings as in the main campaign. The velocity tasks are trained for $24{,}000$ steps and the navigation task SafeCarButton1 for $16{,}000$ steps, with evaluation every $2{,}000$ and $1{,}000$ steps respectively. The temperature is varied over $\eta \in \{0.1, 0.5, 1.0, 2.0, 5.0\}$ with the tier-tuned κ held fixed, and the weight over $\kappa \in \{0.0, 0.15, 0.35, 0.7, 1.5\}$ with $\eta = 1.0$ held fixed.