Accelerating Safe Reinforcement Learning with Massive Parallelism

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

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

Abstract

We investigate the problem of scaling safe reinforcement learning (safe RL) to massively parallel training regimes, where a large number of environments are executed with short rollout horizons. Although this setting improves data throughput and wall-clock efficiency, it creates a mismatch with constrained Markov decision processes (CMDPs), whose safety constraints are defined over full-episode cost returns. We address this mismatch by analyzing staggered environment resets as a phase-mixture distribution over short trajectory segments. This analysis characterizes the bias induced by phase coverage and motivates a phase-aggregated cost estimator that reconstructs episode-level cost estimates from short rollout batches. We further incorporate finite-sample uncertainty and distribution mismatch into a constraint-tightening framework, yielding a scalable Safe RL method compatible with on-policy optimization. Using an MJX-based implementation of Safety Gym, our experiments show that the proposed method achieves performance comparable to non-massively parallel implementations while substantially reducing wall-clock training time. Code is available at: `GitHub Repository`.

1 Introduction

Reinforcement learning (RL) provides a general framework for learning decision-making policies through sequential interaction with an environment and is commonly formulated as a Markov decision process (MDP). In many real-world applications, however, policies must satisfy safety requirements such as collision avoidance, energy limits, and operational constraints [7]. Safe RL addresses this challenge through constrained MDPs (CMDPs), where policy optimization is performed under explicit constraints [3].

Recent advances in accelerator hardware have enabled massively parallel simulation, where simulation and learning can be executed with high throughput on devices such as GPUs and TPUs [11]. This regime is particularly beneficial for on-policy RL, which requires data collected from the current policy for each update and has traditionally suffered from limited sample efficiency. By collecting short rollouts concurrently from many environments, massively parallel training enables frequent policy updates with large effective batches [16]. It also substantially reduces wall-clock training time, making repeated runs for statistical evaluation and rapid parameter tuning feasible within a limited

*Corresponding author.

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

experimental budget [2, 5]. Despite these advantages, Safe RL has not fully incorporated this training regime, and existing methods remain misaligned with its data-processing characteristics.

A key challenge in applying massively parallel training to Safe RL is estimating reliable cost returns for policy update. In CMDPs, constraints are typically formulated in terms of cumulative costs, and existing Safe RL implementations commonly rely on episodic cost returns from completed or terminated episodes to balance reward maximization and constraint satisfaction [7, 18]. However, updates in massively parallel training are highly likely to occur without any episode terminating in any of the parallel environments, particularly in tasks without early termination [4]. Even when an episode terminates, the cost return of the full trajectory is biased by earlier policies in two ways: **P1** the cost return includes costs incurred by actions taken under earlier policies, and **P2** these actions influence the initial-state distributions of subsequent rollouts.

In this paper, we first revisit staggered reset [4], originally introduced to address the cyclical nonstationarity problem. By initializing parallel environments at different episode time steps, each update batch covers the entire episode horizon through short rollouts from different environments. This enables cost estimation by aggregating short rollout segments from different phases, each generated entirely by the current policy, which can address **P1**. We further analyze how policy history biases the phase-initial state distribution and apply tightening to handle **P2**. The resulting framework enables existing Safe RL algorithms to train in the massively parallel regime, improving wall-clock efficiency while maintaining reliable constraint satisfaction. The main contributions are summarized as follows:

- We analyze staggered reset as a phase-mixture batch collection scheme, characterize how policy history biases cost returns and derive computable bounds for the return discrepancy.
- We propose a massively parallel safe-RL framework with staggered reset that estimates cost directly from the phase-mixture batch and instantiate it with phase-initial state distribution correction estimation (PIS-DICE), a phase-initial density-ratio estimator for rollout-start state distributions.
- We provide an implementation of Safety Gym exploration tasks using MuJoCo XLA (MJX) for end-to-end accelerator side training and evaluation, showing improved wall-clock efficiency with reliable cost control.

2 Preliminaries

2.1 Constrained Markov Decision Processes

A CMDP is defined by $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, d_0, r, c, \gamma, b_c)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P(s' \mid s, a)$ is the transition probability, d_0 is the initial state distribution, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $c : \mathcal{S} \times \mathcal{A} \to [0, c_{\max}]$ is the cost function, $\gamma \in [0, 1)$ is the discount factor, and b_c is the cost limit. Under a stationary policy π, the discounted CMDP objective is defined as follows [7, 18]:

$$\max_{\pi} J_r(\pi) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] \quad \text{s.t.} \quad J_c(\pi) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t c_t\right] \leq b_c. \tag{1}$$

Assumption 2.1 (Bounded rewards, costs, and returns)**.** For $f \in \{r, c\}$, there exists $f_{\max} < \infty$ such that $|f(s, a)| \leq f_{\max}$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$.

For discounted infinite-horizon returns, the bound is $B_f = f_{\max}/(1 - \gamma)$. For finite-horizon returns, the bound is $B_f = f_{\max} \sum_{t=0}^{H-1} \gamma^t$, with $\gamma = 1$ allowed for undiscounted episodic returns. The cost limit b_c must be expressed on the same return scale as J_c.

Although we treat discounted returns, many safe RL tasks set the cost threshold as an undiscounted episodic sum. To support both, the return bound is $B_f = f_{\max}/(1 - \gamma)$ for the discounted infinite-horizon case and $B_f = f_{\max} \sum_{t=0}^{H-1} \gamma^t$ for a finite horizon H.

2.2 Massively parallel data collection under staggered reset

In massively parallel on-policy training, each policy update uses length-K rollouts from E parallel environments. We consider episodic tasks with a fixed horizon H, so the discounted returns in Eq. (1)

truncate at H. When $K \ll H$, synchronous reset concentrates each batch on a narrow part of the task horizon, inducing a repeating distribution shift known as cyclical nonstationarity [16, 11, 4]. Staggered reset mitigates this by starting groups of environments at different episode steps, so that a single batch covers the whole horizon rather than a narrow slice.

On-policy RL has traditionally updated frequently from relatively small batches, whereas massively parallel simulation collects short rollouts across a large number of environments and updates from a much larger batch at far lower wall-clock cost. Safe RL fits this regime naturally, since its on-policy algorithms already rely on large batches to estimate the cost constraint with low variance. Massively parallel safe RL has been demonstrated on Isaac Gym [12], but it has not become widespread and has not been studied from an algorithm training perspective, in part because the available tasks have short horizons which avoid the short rollout regime ($K \ll H$).

2.3 Distribution correction from flow identities

Distribution correction is commonly used in off-policy and offline RL to correct the mismatch between the distribution induced by a target policy and the distribution represented by sampled transitions [14, 22, 23]. For the discounted infinite-horizon formulation, let $d^\pi(s,a)$ denote the normalized discounted state–action occupancy distribution induced by a stationary policy π, and let $d^D(s,a)$ denote the underlying state–action sampling distribution from which the transitions in D are collected.

Assumption 2.2 (Absolute continuity)**.** d^π is absolutely continuous with respect to d^D.

With Assumption 2.2, the density ratio is defined as $\rho^{\pi/D}(s,a) = \frac{d^\pi(s,a)}{d^D(s,a)}$. Multiplying by $\rho^{\pi/D}$ turns an expectation under d^D into an expectation under d^π, so for a bounded function f,

$$\mathbb{E}_{(s,a)\sim d^\pi}[f(s,a)] = \mathbb{E}_{(s,a)\sim d^D}\left[\rho^{\pi/D}(s,a) f(s,a)\right]. \tag{2}$$

However, neither density is available in closed form and d^D is accessed only through samples, so $\rho^{\pi/D}$ cannot be computed directly. Instead, under a stationary policy π, the discounted state-action distribution d^π satisfies the flow identity

$$d^\pi(s',a') = (1-\gamma) d_0(s')\pi(a' \mid s') + \gamma\, \pi(a' \mid s') \sum_{s,a} d^\pi(s,a) P(s' \mid s,a). \tag{3}$$

The flow into (s',a') splits into the initial-state contribution and the transition flow from previous state-action pairs. Substituting $d^\pi = \rho^{\pi/D} d^D$ into Eq. (3) leaves $\rho^{\pi/D}$ as the only unknown, so distribution correction estimation (DICE) methods [14, 22, 23] estimate $\rho^{\pi/D}$ by enforcing Eq. (3) on samples of D, without estimating the two densities separately.

3 Methodology

We formalize constraint estimation from phase-mixture batches collected from environments with different phases. We first analyze the return computed from the collected batch and compare it with the corresponding current policy return. We then construct phase-wise cost surrogates and tighten the update by considering finite-sample error and distribution mismatch. Finally, we specialize the construction to uniform staggered reset and introduce PIS-DICE as the distribution correction estimator for phase-initial states.

3.1 Phase-mixture return analysis

When training with parallel environments and short rollouts, early termination of individual environments or staggered reset distributes environments over phases $\phi \in \{0, \dots, H-1\}$ [2, 4]. At epoch n, define the phase distribution by $q_n(\phi) = \frac{1}{E}|\{e : \phi_e = \phi\}|$, and let $E_\phi = E q_n(\phi)$ be the number of environments in phase ϕ. Let $d_{n,\phi}$ be the collected phase-initial state distribution [21, 8, 20]. Under policy π_n, define $\zeta_{n,\phi,k}(s,a) = (P^k_{\pi_n} d_{n,\phi})(s)\pi_n(a|s)$. For any return function $f \in \{r, c\}$, set $\ell_{f,n}(\phi,k) = \mathbb{E}_{(s,a)\sim\zeta_{n,\phi,k}}[f(s,a)]$. The expected K-step return from rollouts initialized at phase ϕ

is

$$j^K_{f,n}(\phi) = \sum_{k=0}^{K-1} \gamma^k \ell_{f,n}(\phi, k). \tag{4}$$

For the collected batch, the empirical K-step phase-mixture return is

$$\widehat{j}^K_{f,n} = \frac{1}{E} \sum_{e=1}^{E} \sum_{k=0}^{K-1} \gamma^k f(s_{e,\phi_e+k}, a_{e,\phi_e+k}). \tag{5}$$

Lemma 3.1. *For a batch with phase distribution q_n, the empirical K-step return is an unbiased estimator of the phase-mixture return, given by*

$$\mathbb{E}\left[\widehat{j}^K_{f,n}\right] = \sum_{\phi=0}^{H-1} q_n(\phi) j^K_{f,n}(\phi). \tag{6}$$

Proof. See Appendix A.1.1. □

The collected phase-initial distribution $d_{n,\phi}$ may be shaped by previous policies, which changes the state–action distribution $\zeta_{n,\phi,k}$ and the expected return $\ell_{f,n}(\phi, k)$. Let $\bar{d}_{n,\phi} = P^{\phi}_{\pi_n} d_0$ denote the phase-initial distribution induced by the current policy π_n. The corresponding current policy state–action distribution is $\bar{\zeta}_{n,\phi,k}(s, a) = \left(P^k_{\pi_n} \bar{d}_{n,\phi}\right)(s)\pi_n(a|s)$. The current policy one-step return is $\bar{\ell}_{f,n}(\phi, k) = \mathbb{E}_{(s,a)\sim\bar{\zeta}_{n,\phi,k}}[f(s, a)]$, and the corresponding K-step phase-mixture return is

$$\bar{j}^K_{f,n} = \sum_{\phi=0}^{H-1} q_n(\phi) \sum_{k=0}^{K-1} \gamma^k \bar{\ell}_{f,n}(\phi, k). \tag{7}$$

Proposition 3.2 (Initial phase state bias bound)**.** *Under Assumption 2.1, for $f \in \{r, c\}$, the initial phase state mismatch bias can be bounded as*

$$\left|\mathbb{E}\left[\widehat{j}^K_{f,n}\right] - \bar{j}^K_{f,n}\right| \le 2 f_{\max} \sum_{\phi=0}^{H-1} q_n(\phi) \sum_{k=0}^{K-1} \gamma^k \left\|\zeta_{n,\phi,k} - \bar{\zeta}_{n,\phi,k}\right\|_{\mathrm{TV}}. \tag{8}$$

Proof. See Appendix A.1.2. □

Proposition 3.2 identifies the distributional term separating the phase-mixture return estimated from the batch and the corresponding current policy phase-mixture return. Even in on-policy algorithms, this correction is required when the collected phase-initial distribution $d_{n,\phi}$ differs from the current policy phase-initial distribution $\bar{d}_{n,\phi}$.

3.2 Phase-wise cost surrogates and tightening

Since safe RL algorithms need cost returns for dual updates or feasibility recovery, we consider two cost surrogates from the collected phase-mixture batch. The sampled cost surrogate aggregates the sampled cost $c(s, a)$ with phase-coverage correction. The critic value surrogate evaluates the cost critic $V_{c,n}(s)$ on states sampled from the same phase-mixture batch. Define the binary indicator $I_{\phi,k}(t) = \mathbb{I}\{t = \phi + k\}$. The coverage mass of timestep t and the coverage-correction weight are

$$m_{q_n}(t) = \sum_{\phi=0}^{H-1} q_n(\phi) \sum_{k=0}^{K-1} I_{\phi,k}(t), \qquad w_{q_n}(\phi, k) = \frac{\gamma^{\phi+k}}{m_{q_n}(\phi + k)}. \tag{9}$$

Assume $m_{q_n}(t) > 0$ for $t = 0, \ldots, H-1$, and let $\Phi_n := \{\phi : q_n(\phi) > 0\}$ denote the set of observed phases. The two phase-wise surrogates are

$$\begin{aligned}
\widehat{J}^{K,\mathrm{sample}}_{c,n} &= \sum_{\phi\in\Phi_n} q_n(\phi) \frac{1}{E_\phi} \sum_{e:\phi_e=\phi} \sum_{k=0}^{K-1} w_{q_n}(\phi, k) c(s_{e,\phi_e+k}, a_{e,\phi_e+k}), \\
\widehat{J}^{K,\mathrm{value}}_{c,n} &= \sum_{\phi\in\Phi_n} q_n(\phi) \frac{1}{E_\phi K} \sum_{e:\phi_e=\phi} \sum_{k=0}^{K-1} V_{c,n}(s_{e,\phi_e+k}).
\end{aligned} \tag{10}$$

Here $V_{c,n}(s_t)$ predicts $\mathbb{E}_{\pi_n}[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} c(s_\tau, a_\tau) \mid s_t]$.

Proposition 3.3 (Finite-sample error bound)**.** *Under Assumption 2.1, choose either surrogate in Eq.* (10) *and write it as* $\widehat{J}_{c,n}^K = \frac{1}{E}\sum_{e=1}^{E} X_e$*, where the* X_e *are independent per-environment random variables. Let* R_n *be a deterministic range satisfying* $0 \le X_e \le R_n$ *for every environment* e*, and let* $S^2 = \frac{1}{E-1}\sum_{e=1}^{E}(X_e - \widehat{J}_{c,n}^K)^2$*. By the one-sided empirical Bernstein inequality, the finite-sample error of the cost surrogate is bounded with probability at least* $1-\delta$*,*

$$\mathbb{E}\left[\widehat{J}_{c,n}^K\right] - \widehat{J}_{c,n}^K \le \sqrt{\frac{2S^2\log(2/\delta)}{E}} + \frac{7R_n\log(2/\delta)}{3(E-1)}. \tag{11}$$

Proof. See Appendix A.1.3. □

Theorem 3.4 (Constraint satisfaction with tightened cost surrogate)**.** *Assume that the selected surrogate scale satisfies the distribution mismatch upper bound* $J_c(\pi_n) \le \mathbb{E}[\widehat{J}_{c,n}^K] + \varepsilon_n$*. Together with Propositions 3.2 and 3.3, if the cost surrogate from the phase-mixture batch satisfies*

$$\widehat{J}_{c,n}^K + \beta_n + \varepsilon_n \le b_c, \tag{12}$$

then $J_c(\pi_n) \le b_c$ *with probability at least* $1-\delta$*.*

Proof. See Appendix A.1.4. □

3.3 Uniform staggered reset

Massively parallel environment training is typically executed under JIT/XLA compilation [11, 16], where keeping rollout tensors shape stable after compilation is important. Therefore, staggered reset partitions the E environments into M phase groups and assigns each group a fixed phase offset mK for $m = 0, \ldots, M-1$ along the task horizon $H = MK$. Environments that terminate early are kept in a masked terminal state until the group reset point or the episode end point, so no individual environment reset is performed. This group-wise distribution yields a uniform phase allocation over non-overlapping horizon segments, with mass $1/M$ on each observed phase and no mass on other phases, and provides a practical instance of the phase-mixture analysis in Section 3.1.

Theorem 3.5 (Uniform staggered reset induces a Cesàro phase mixture)**.** *Under uniform staggered reset, the phase-mixture return in Eq.* (6) *is the finite Cesàro mean of the phase-wise returns* $j_{f,n}^K(\phi)$ *from Eq.* (4)*, namely* $\frac{1}{M}\sum_{m=0}^{M-1} j_{f,n}^K(mK)$*.*

Proof. See Appendix A.1.5. □

Corollary 3.6 (Cesàro phase-mixture return)**.** *Under uniform staggered reset, the phase-mixture return in Lemma 3.1 becomes*

$$\mathbb{E}\left[\widehat{j}_{f,n}^K\right] = \frac{1}{M}\sum_{m=0}^{M-1} j_{f,n}^K(mK). \tag{13}$$

Corollary 3.7 (Cesàro initial phase state bias bound)**.** *Under uniform staggered reset and Assumption 2.1, the mismatch bound in Proposition 3.2 becomes*

$$\left|\mathbb{E}\left[\widehat{j}_{f,n}^K\right] - \bar{j}_{f,n}^K\right| \le \frac{2f_{\max}}{M}\sum_{m=0}^{M-1}\sum_{k=0}^{K-1}\gamma^k \left\|\zeta_{n,mK,k} - \bar{\zeta}_{n,mK,k}\right\|_{\mathrm{TV}}. \tag{14}$$

Corollary 3.8 (Uniform phase-wise cost surrogate under staggered reset)**.** *Under uniform staggered reset, the phase-wise surrogate in Eq.* (10) *becomes*

$$\begin{aligned}
\widehat{J}_{c,n}^{K,\mathrm{sample}} &= \sum_{m=0}^{M-1}\frac{1}{E_{mK}}\sum_{e:\phi_e=mK}\sum_{k=0}^{K-1}\gamma^{mK+k}c(s_{e,mK+k}, a_{e,mK+k}),\\
\widehat{J}_{c,n}^{K,\mathrm{value}} &= \frac{1}{M}\sum_{m=0}^{M-1}\frac{1}{E_{mK}K}\sum_{e:\phi_e=mK}\sum_{k=0}^{K-1}V_{c,n}(s_{e,mK+k}).
\end{aligned} \tag{15}$$

The corollary proofs are given in Appendices A.1.6, A.1.7, and A.1.8, respectively.

3.4 PIS-DICE: phase-initial state distribution correction

The TV term in Eq. (14) compares collected state–action distributions with current policy state–action distributions at each phase and rollout offset, which cannot be estimated directly in the short-rollout training regime. Therefore, we extend the concept of distribution correction from state–action distributions to phase-initial state distributions.

Under uniform staggered reset, a phase-initial pair is $x = (s, \phi)$, where s is the state at the start of phase ϕ. Let $\mathcal{K}_n(x^+ \mid x)$ be the transition kernel obtained after executing π_n for K environment steps. For consecutive phase-initial pairs used by the estimator, $x^+ = (s^+, \phi + K)$ with $\phi + K < H$. For every observed phase, assume $\bar{d}_{n,\phi}$ is absolutely continuous with respect to $d_{n,\phi}$. Writing their densities with the same symbols, define the phase-initial correction weight $\bar{\rho}_n(s, \phi) = \frac{\bar{d}_{n,\phi}(s)}{d_{n,\phi}(s)}$, so $\mathbb{E}_{s\sim d_{n,\phi}}[\bar{\rho}_n(s, \phi)]$=1. For each interior phase $m = 0, \ldots, M-2$, the phase-initial one-block relation is

$$\mathbb{E}_{s\sim d_{n,mK},\, x^+\sim\mathcal{K}_n(\cdot|(s,mK))}\left[\bar{\rho}_n(s, mK)h(x^+)\right] = \mathbb{E}_{s\sim\bar{d}_{n,(m+1)K}}\left[h(s, (m+1)K)\right], \qquad \forall h \in \mathcal{H}. \tag{16}$$

Here $\mathcal{H}$ is a class of bounded measurable functions. The sampled one-block residual below is used as an empirical fitting objective for the learned weights, while the formal bounds use the change-of-measure identity directly. For a fixed phase ϕ, let $j^K_{c,n,\phi}(s)$ be the K-step expected cost return under the current policy from phase state s. Averaging this conditional expected return over the collected and current policy phase-initial state distributions gives $C^K_{n,\phi}$=$\mathbb{E}_{s\sim d_{n,\phi}}[j^K_{c,n,\phi}(s)]$ and $\bar{C}^K_{n,\phi}$=$\mathbb{E}_{s\sim\bar{d}_{n,\phi}}[j^K_{c,n,\phi}(s)]$. The phase-initial change of measure gives $\bar{C}^K_{n,\phi} - C^K_{n,\phi}$=$\mathbb{E}_{s\sim d_{n,\phi}}[(\bar{\rho}_n(s, \phi) - 1)j^K_{c,n,\phi}(s)]$. Thus, we use this phase-initial correction to handle the TV term in Eq. (14).

PIS-DICE uses two learned models. The ratio model outputs a nonnegative phase-initial correction weight. The residual model maps a phase-initial pair to a scalar whose difference across one rollout block defines an auxiliary fitting objective for the ratio model. In the implementation, the ratio-model output is $\rho_n(s, \phi) = \epsilon_\rho + \mathrm{softplus}(u_\eta(s, \phi/(H-1)))$, with $\epsilon_\rho > 0$. Since the exact ratio $\bar{\rho}_n(s, \phi)$ is not observed, the implementation fits ρ_n using collected phase-initial samples and the states reached after the rollout block. The residual model is used only for this empirical fitting objective. The theoretical statements below are stated for the exact ratio $\bar{\rho}_n$, and Proposition 3.10 quantifies the additional error introduced by replacing $\bar{\rho}_n$ with learned weights.

For each environment e, x_e denotes the phase-initial pair and x_e^+ denotes the pair obtained after the collected rollout block. Let f_ω be the residual-model output. The sampled residual is $\Delta_n(\eta, \omega) = E^{-1}\sum_{e=1}^E \rho_n(x_e)(f_\omega(x_e) - f_\omega(x_e^+))$. The residual model is trained to expose this discrepancy, and the ratio model is trained to reduce the exposed discrepancy together with phase-wise normalization and ratio regularization. Let $\mathrm{sg}(\cdot)$ denote stop-gradient. The losses are

$$\begin{aligned}
\mathcal{L}_{\rho,n}(\eta) &= \frac{1}{E}\sum_{e=1}^E \rho_n(x_e)\,\mathrm{sg}\big[f_\omega(x_e) - f_\omega(x_e^+)\big] \\
&\quad + \lambda_{\mathrm{norm}}\mathcal{R}_{\mathrm{norm},n}(\rho_n) + \lambda_{\rho 2}\frac{1}{E}\sum_{e=1}^E(\rho_n(x_e) - 1)^2, \\
\mathcal{L}_{f,n}(\omega) &= -\frac{1}{E}\sum_{e=1}^E \mathrm{sg}[\rho_n(x_e)]\left(f_\omega(x_e) - f_\omega(x_e^+)\right) + \lambda_f\frac{1}{E}\sum_{e=1}^E f_\omega(x_e)^2,
\end{aligned} \tag{17}$$

where $\mathcal{R}_{\mathrm{norm},n}$ is the phase-wise normalization term, $\mathcal{R}_{\mathrm{norm},n}(\rho)$=$\sum_{\phi\in\Phi_n}(|I_\phi|^{-1}\sum_{e\in I_\phi}\rho(x_e) - 1)^2$. After training, the raw weights $\rho_n(x_e)$ are clipped to $[1/\rho_{\max}, \rho_{\max}]$ and normalized within each phase. We write the final normalized weight as $\tilde{\rho}_{n,e} = \frac{|I_{\phi_e}|\rho^{\mathrm{clip}}_{n,e}}{\sum_{j\in I_{\phi_e}}\rho^{\mathrm{clip}}_{n,j}}$, where $\rho^{\mathrm{clip}}_{n,e} = \mathrm{clip}(\rho_n(x_e), 1/\rho_{\max}, \rho_{\max})$, so $|I_\phi|^{-1}\sum_{e\in I_\phi}\tilde{\rho}_{n,e} = 1$ for every observed phase. The losses above are used only to fit $\tilde{\rho}_n$. The formal bounds below use the exact change-of-measure identity for $\bar{\rho}_n$ and then account for replacing $\bar{\rho}_n$ with $\tilde{\rho}_n$ through an explicit learned-ratio error term.

Proposition 3.9 (PIS-DICE phase-initial discrepancy upper bound)**.** *With the exact phase-initial ratio $\bar{\rho}_n$ defined above, under Assumption 2.1, $0 \le j^K_{c,n,\phi}(s) \le B_c$, the phase-wise cost discrepancy*

satisfies

$$\left|\frac{1}{M}\sum_{m=0}^{M-1}\left(\bar{C}^K_{n,mK}-C^K_{n,mK}\right)\right| \le \frac{B_c}{M}\sum_{m=0}^{M-1}\mathbb{E}_{s\sim d_{n,mK}}[|\bar{\rho}_n(s,mK)-1|]\,. \tag{18}$$

Proof. See Appendix A.1.9. □

Proposition 3.9 gives a phase-initial change-of-measure bound for the cost discrepancy induced by initial phase-state mismatch. The bound replaces the state–action TV terms over all phase offsets with a phase-initial correction term. The practical correction therefore depends on the accuracy of the learned ratio.

Proposition 3.10 (Learned PIS-DICE correction error)**.** *For the exact phase-initial ratio $\bar{\rho}_n$, condition on a learned nonnegative estimate $\tilde{\rho}_n$. Assume the selected surrogate cost scalar $Y_n(s,\phi)$ satisfies $|Y_n(s,\phi)| \le B_Y$. Then*

$$\begin{aligned}&\left|\frac{1}{M}\sum_{m=0}^{M-1}\mathbb{E}_{s\sim\bar{d}_{n,mK}}[Y_n(s,mK)]-\frac{1}{M}\sum_{m=0}^{M-1}\mathbb{E}_{s\sim d_{n,mK}}[\tilde{\rho}_n(s,mK)Y_n(s,mK)]\right| \\ &\quad\le \frac{B_Y}{M}\sum_{m=0}^{M-1}\mathbb{E}_{s\sim d_{n,mK}}[|\bar{\rho}_n(s,mK)-\tilde{\rho}_n(s,mK)|]\,.\end{aligned} \tag{19}$$

Proof. See Appendix A.1.10. □

3.5 Practical update rule

The implementation uses the uniform staggered reset form in Eq. (15). The sampled cost form accumulates binary costs from the phase-mixture batch and compares the result with b_c on the task cost-return scale. The critic value form evaluates the same batch with $V_{c,n}$ and compares it with the cost limit on the critic-return scale.

For the selected cost surrogate, write the per-environment scalar as $Y_{n,e}$. For the sampled cost form, $Y_{n,e}{=}M\sum_{k=0}^{K-1}\gamma^{\phi_e+k}c(s_{e,\phi_e+k},a_{e,\phi_e+k})$; for the critic value form, $Y_{n,e}{=}K^{-1}\sum_{k=0}^{K-1}V_{c,n}(s_{e,\phi_e+k})$. The finite-sample term β_n is computed from Proposition 3.3 for the selected scalar $Y_{n,e}$.

PIS-DICE provides the normalized learned weight $\tilde{\rho}_{n,e}$. The update keeps the unweighted cost surrogate and uses $D_{n,e}{=}(\tilde{\rho}_{n,e}-1)Y_{n,e}$ to compute the mismatch correction

$$\varepsilon_n^{\mathrm{PIS}} = \left[\frac{1}{E}\sum_{e=1}^{E}D_{n,e}+\sqrt{\frac{2S_D^2\log(1/\delta)}{E}}\right]_+, \quad S_D^2=\frac{1}{E-1}\sum_{e=1}^{E}\left(D_{n,e}-\frac{1}{E}\sum_{j=1}^{E}D_{n,j}\right)^2.$$

The policy update uses $\varepsilon_n=\varepsilon_n^{\mathrm{PIS}}$ in Eq. (12), combining the finite-sample term β_n with the empirical correction computed from the learned weights. In the main experiments, PIS-DICE changes only the cost-side tightening, while the actor objective remains unchanged.

4 Evaluation

We focus on evaluating whether the proposed massively parallel safe RL training framework can maintain reliable cost control with short, accelerator-friendly updates. The experiments first evaluate the effect of uniform staggered reset and the additional effect of tightening. We then compare the massively parallel training approach with the small-vectorized environment setting used in existing implementations in terms of both sample efficiency measured by total environment steps and wall-clock efficiency.

Table 1: Evaluation of CPO across 2k, 4k, and 8k parallel environments with a fixed rollout length of 10. Values are mean ± standard deviation of the last 10 epochs over eight seeds. Shading indicates violation of the cost threshold (cost > 25). **Bold** highlights, for each environment count, the highest reward that satisfies the cost threshold.

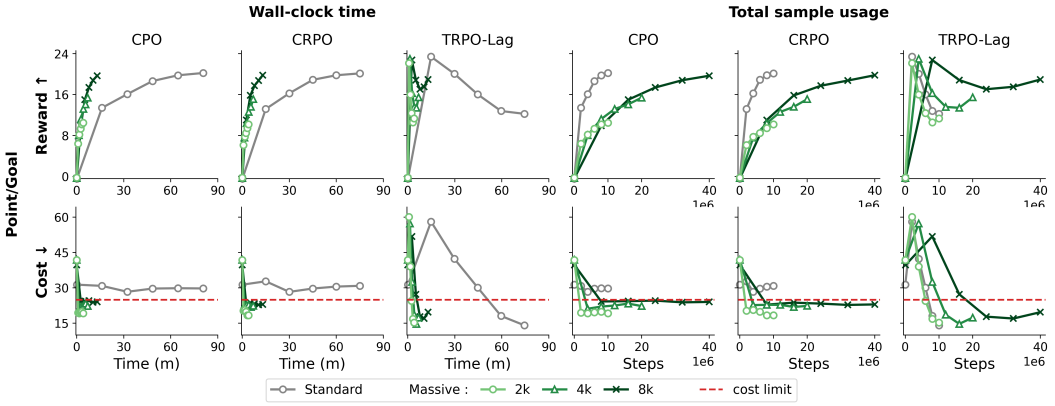
Init	Cost est.	Env	Point/Goal				Car/Goal			
			w/o tightening		w/ tightening		w/o tightening		w/ tightening	
			Reward	Cost	Reward	Cost	Reward	Cost	Reward	Cost
Standard	LR	2K	15.13±9.94	55.48±24.54	–	–	29.14±4.76	65.87±12.06	–	–
Staggered	SC	2K	14.57±2.78	27.08±0.84	7.56±3.06	13.88±1.59	25.12±2.11	28.63±1.46	18.52±2.83	14.38±1.18
		4K	17.28±5.37	26.27±1.23	13.42±3.52	20.00±3.49	27.57±1.76	26.87±1.18	**24.38**±2.04	**18.43**±0.75
		8K	21.23±1.05	27.02±1.50	**18.83**±0.73	**21.19**±2.05	32.17±0.95	26.52±1.38	**29.57**±0.70	**21.06**±0.70
	CV	2K	15.29±3.76	29.55±3.43	**13.42**±2.06	**24.53**±3.93	26.13±2.57	28.80±3.16	**22.69**±3.00	**22.19**±3.23
		4K	19.30±2.27	29.61±3.04	**17.40**±2.23	**24.57**±2.98	30.16±2.04	29.87±3.98	28.46±1.37	25.78±3.53
		8K	21.46±1.17	27.82±2.25	20.55±2.29	26.91±3.21	32.67±0.86	29.75±2.06	31.21±1.40	26.37±3.17

Setup We implement Goal tasks with Point and Car agents in MJX, following Safety Gym-style navigation task definitions [15, 9] and the accelerator-side simulation design of JAX-based physics engines [6]. We further implement a locomotion task with a Laikago robot [10] in both MJX and MuJoCo Warp. To compare different constraint-handling paradigms, CPO, TRPO-Lag, and CRPO were chosen as representative primal-dual, Lagrangian, and primal methods, respectively [1, 17, 19]. By default, the massively parallel setting uses 2,000 environments collecting a batch size of 20,000 steps per update over 500 epochs. All experiments on the Goal tasks were conducted on Google TPU v4, while locomotion tasks were conducted on machine with NVIDIA GeForce RTX 4090 GPU.

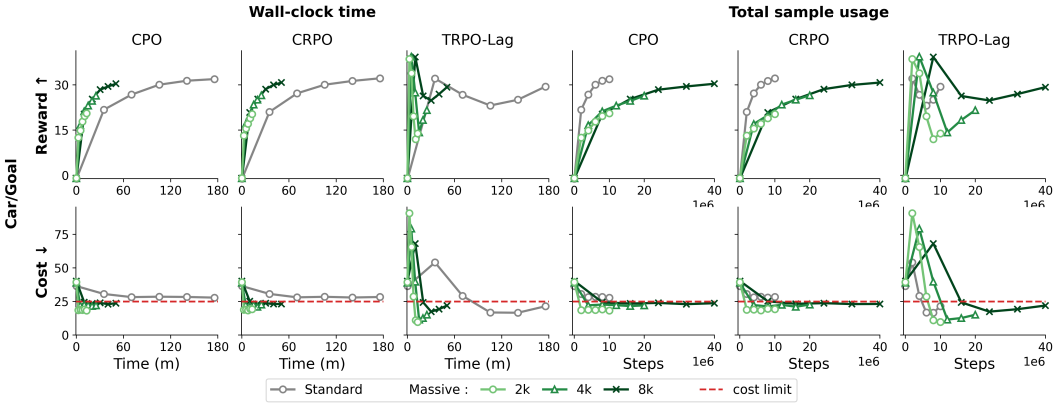
Effect of staggered reset and tightening. We first compare massively parallel training with short-rollout updates using the two initialization schemes, standard synchronous reset and staggered reset, and then evaluate the additional effect of the tightening method. Since standard initialization does not provide a finished-episode cost return at every update, the latest completed-episode cost return is reused for updates without completed episodes; we denote this estimate by LR. Staggered reset uses two cost surrogates, denoted as SC for sampled cost and CV for critic value estimates, evaluated with and without tightening. We further scale parallelism from 2k to 4k and 8k environments, so the batch size increases to 40,000 and 80,000 transitions, respectively.

Table 1 shows the results of CPO on the Goal task across different environment scales. At 2k environments, standard initialization exceeds the cost limit, with completed episode cost returns available only sparsely. Staggered reset reduces cost returns by providing estimates from every phase-mixture batch. Without tightening, both surrogates still exceed the cost limit, even as increasing parallelism leads to higher rewards. Applying tightening further reduces costs at the expense of reward: SC keeps the reported mean episodic cost below the limit across the tested scales, whereas CV exceeds it in some configurations. The reward gap between tightened and untightened training narrows at larger scales, consistent with the smaller finite-sample bound afforded by increased sampling.

Comparison with non-massive environment training. We compare the proposed massively parallel setting against the small-vectorized environment setting used in existing implementations, denoted as the standard setting. Figure 1 shows the learning curves of the Goal tasks in terms of wall-clock time and total environment steps. With the same number of environment steps, the massively parallel setting generally converges to lower rewards than the standard setting using full task-horizon rollouts, which achieves better sample efficiency by providing complete task trajectories. However, the primary advantage of massively parallel training lies in wall-clock efficiency rather than sample efficiency. By processing many short rollouts from different phases concurrently on accelerators, the massively parallel setting increases transition throughput and enables more frequent policy updates within the same elapsed time. In terms of wall-clock time, training in the massively parallel setting converges to higher reward levels across algorithms as parallelism increases. In



(a) Point/Goal. The left side shows the wall-clock wise comparison and the right side shows the total sample wise comparison.



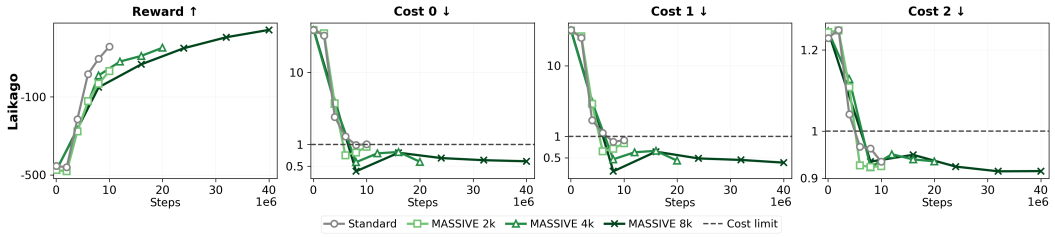
(b) Car/Goal. The left side shows the wall-clock wise comparison and the right side shows the total sample wise comparison.

Figure 1: Wall-clock wise and total sample wise learning curves of all algorithms on the Goal task. Columns correspond to CPO, CRPO, and TRPO-Lag. Markers are shown every 100 epochs and report trailing 10-epoch averages, and the horizontal line marks the cost limit.

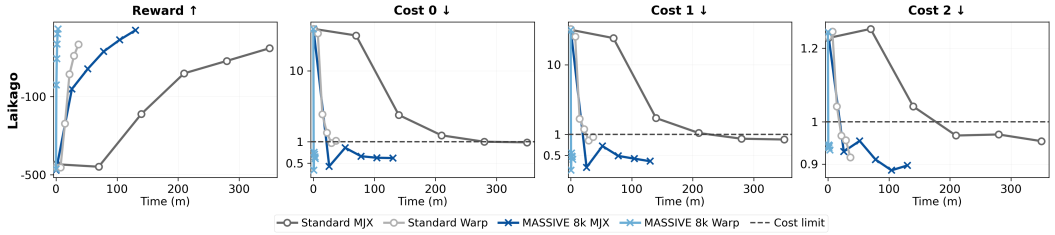
particular, 8k configuration achieves rewards comparable to the standard setting with better constraint satisfaction in substantially less training time.

Figure 2 shows the learning curves of the Laikago locomotion task implemented using MJX and MuJoCo Warp. Consistent with the trends observed in the Goal tasks, reward convergence improves as parallelism increases, reaching a level comparable to the standard setting at 4K environments and surpassing it at 8K. In terms of wall-clock time, both MJX and MuJoCo Warp achieve substantially shorter training times at 8K environments than the standard setting. The difference between the two implementations further suggests that additional efficiency gains can be obtained through the choice of GPU simulation backend. Overall, these results demonstrate that massively parallel training could provide substantial wall-clock efficiency gains for safe RL research while maintaining or improving performance.

Ablation studies We conduct ablation studies to compare the effects of increasing rollout length against increasing the number of parallel environments and to examine how each tightening term affects training. The results are provided in Appendix B.



(a) Comparison across environment scales in terms of total environment steps, averaged over MJX and MuJoCo Warp.



(b) Comparison of the standard and 8k environment settings in terms of wall-clock time using MJX and MuJoCo Warp.

Figure 2: Learning curves of CRPO on the Laikago locomotion task. Markers are shown every 100 epochs and report trailing 10-epoch averages. Costs are normalized by their respective limits, with horizontal lines indicating the normalized cost limit of 1.

5 Limitations

The theoretical results assume bounded costs, independent samples across environments, sufficient phase coverage, and uniform staggered resets for the simplified corollaries. Freezing terminated environments wastes some computation but preserves fixed computation shapes for efficient accelerator execution, allowing the throughput gains from massive parallelism to potentially outweigh this overhead. Nevertheless, extending the implementation to arbitrary asynchronous or highly nonuniform reset schedules remains for future work. The practical tightening rule combines finite-sample tightening with density ratio correction, but the ablations show that most of the improvement in constraint satisfaction comes from the finite-sample term. Further evaluation on complex tasks with greater phase-initial distribution mismatch would help clarify the contribution of distribution correction. Applying tightening to multiple constraints may also become overly conservative and restrict learning, motivating further study of alternatives to tightening. The CV surrogate relies on critics trained with bootstrapped k-step targets from short rollouts and shows less consistent cost control than SC in the reported experiments. The current tightening rule does not account for critic estimation errors, and extending it to cover these errors could further increase conservatism. An alternative is to investigate distributional critics that exploit the large training batches to improve cost estimation.

6 Conclusion

We propose a massively parallel Safe RL framework that aligns short-rollout training with episode-level CMDP constraints through phase-covered cost estimation. By analyzing staggered reset as a phase-mixture data collection scheme, we derive cost surrogates with finite-sample and phase-initial distribution corrections, and instantiate the correction with PIS-DICE. Experiments on Safety Gym-style Goal tasks implemented in MJX and Laikago locomotion tasks implemented in both MJX and MuJoCo Warp show that the proposed framework improves wall-clock efficiency while keeping costs near their limits. These results demonstrate that safe RL can be effectively extended to massively parallel training regimes, achieving substantial gains in wall-clock efficiency while maintaining constraint satisfaction.

Acknowledgments and Disclosure of Funding

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

This work was also supported with Cloud TPUs from Google's TPU Research Cloud (TRC) program, and we sincerely thank the TRC team for their generous and continuous support.

References

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

[2] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in Neural Information Processing Systems*, 34, 2021.

[3] Eitan Altman. Constrained Markov Decision Processes. Technical Report RR-2574, INRIA, May 1995. URL `https://inria.hal.science/inria-00074109`.

[4] Sid Bharthulwar, Stone Tao, and Hao Su. Staggered environment resets improve massively parallel on-policy reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL `https://openreview.net/forum?id=hesM5BWtOJ`.

[5] Theresa Eimer, Marius Lindauer, and Roberta Raileanu. Hyperparameters in reinforcement learning and how to tune them. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[6] C Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax–a differentiable physics engine for large scale rigid body simulation. *arXiv preprint arXiv:2106.13281*, 2021.

[7] Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, and Alois Knoll. A review of safe reinforcement learning: Methods, theories, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(12):11216–11235, December 2024. ISSN 0162-8828. doi: 10.1109/TPAMI.2024.3457538. URL `https://doi.org/10.1109/TPAMI.2024.3457538`.

[8] Ashish Kumar Jayant and Shalabh Bhatnagar. Model-based safe deep reinforcement learning via a constrained proximal policy optimization algorithm. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

[9] Jiaming Ji, Jiayi Zhou, Borong Zhang, Qingyang Dai, Xuehai Pan, Ruiyang Sun, Weidong Huang, Yiran Geng, and Yaodong Liu. Safety-gymnasium: A unified safe reinforcement learning benchmark. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2023.

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

[11] Sotetsu Koyamada, Shinri Okano, Soichiro Nishimori, Yu Murata, Keigo Habara, Haruka Kita, and Shin Ishii. Pgx: Hardware-accelerated parallel game simulators for reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL `https://openreview.net/forum?id=UvX8QfhfUx`.

[12] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[13] Andreas Maurer and Massimiliano Pontil. Empirical bernstein bounds and sample variance penalization. In *Conference on Learning Theory*, 2009.

[14] Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *Advances in neural information processing systems*, 32, 2019.

[15] Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*, 7(1):2, 2019.

[16] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *5th Annual Conference on Robot Learning*, 2021. URL https://openreview.net/forum?id=wK2fDDJ5VcF.

[17] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, 2015.

[18] Akifumi Wachi, Xun Shen, and Yanan Sui. A survey of constraint formulations in safe reinforcement learning. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, IJCAI '24, 2024. ISBN 978-1-956792-04-1. doi: 10.24963/ijcai.2024/913. URL https://doi.org/10.24963/ijcai.2024/913.

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

[20] Moritz A. Zanger, Karam Daaboul, and J. Marius Zöllner. Safe continuous control with constrained model-based policy optimization. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3512–3519, 2021. doi: 10.1109/IROS51168.2021.9635984.

[21] Sihan Zeng, Thinh T. Doan, and Justin Romberg. Finite-time complexity of online primal-dual natural actor-critic algorithm for constrained markov decision processes. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 4028–4033, 2022. doi: 10.1109/CDC51059.2022.9992837.

[22] Ruiyi Zhang, Bo Dai, Lihong Li, and Dale Schuurmans. Gendice: Generalized offline estimation of stationary values. In *International Conference on Learning Representations*, 2020.

[23] Shangtong Zhang, Bo Liu, and Shimon Whiteson. Gradientdice: Rethinking generalized offline estimation of stationary values. In *International Conference on Machine Learning*, pages 11194–11203, 2020.

A Supplementary proofs and implementation details

A.1 Proofs omitted in the main text

A.1.1 Proof of Lemma 3.1

Proof. Condition on the phase allocation $\{\phi_e\}_{e=1}^{E}$. For each environment e with $\phi_e = \phi$, the distribution of $(s_{e,\phi+k}, a_{e,\phi+k})$ under the rollout policy is $\zeta_{n,\phi,k}$. Hence the definition of $\ell_{f,n}$ gives

$$\mathbb{E}\left[f(s_{e,\phi+k}, a_{e,\phi+k}) \mid \phi_e = \phi\right] = \ell_{f,n}(\phi, k). \tag{20}$$

Starting from the empirical phase-mixture return in Eq. (5) and using linearity of expectation,

$$\mathbb{E}\left[\widehat{j}_{f,n}^{K}\right] = \mathbb{E}\left[\frac{1}{E}\sum_{e=1}^{E}\sum_{k=0}^{K-1}\gamma^k f(s_{e,\phi_e+k}, a_{e,\phi_e+k})\right] \tag{21}$$

$$= \frac{1}{E}\sum_{\phi=0}^{H-1}\sum_{e:\phi_e=\phi}\sum_{k=0}^{K-1}\gamma^k \mathbb{E}\left[f(s_{e,\phi+k}, a_{e,\phi+k}) \mid \phi_e = \phi\right] \tag{22}$$

$$= \frac{1}{E}\sum_{\phi=0}^{H-1}\sum_{e:\phi_e=\phi}\sum_{k=0}^{K-1}\gamma^k \ell_{f,n}(\phi, k) \tag{23}$$

$$= \sum_{\phi=0}^{H-1}\frac{E_\phi}{E}\sum_{k=0}^{K-1}\gamma^k \ell_{f,n}(\phi, k). \tag{24}$$

By the definition $q_n(\phi) = E_\phi/E$ and the phase return in Eq. (4), the final line equals

$$\sum_{\phi=0}^{H-1} q_n(\phi) j_{f,n}^{K}(\phi), \tag{25}$$

which is Eq. (6). □

A.1.2 Proof of Proposition 3.2

Proof. Lemma 3.1 gives the collected phase-mixture expectation in Eq. (6). Expanding $j_{f,n}^{K}(\phi)$ by Eq. (4),

$$\mathbb{E}\left[\widehat{j}_{f,n}^{K}\right] = \sum_{\phi=0}^{H-1} q_n(\phi)\sum_{k=0}^{K-1}\gamma^k \mathbb{E}_{(s,a)\sim\zeta_{n,\phi,k}}[f(s,a)]. \tag{26}$$

The current policy phase-mixture return is Eq. (7), namely

$$\bar{j}_{f,n}^{K} = \sum_{\phi=0}^{H-1} q_n(\phi)\sum_{k=0}^{K-1}\gamma^k \mathbb{E}_{(s,a)\sim\bar{\zeta}_{n,\phi,k}}[f(s,a)]. \tag{27}$$

Subtracting the two displays and applying the triangle inequality yields

$$\left|\mathbb{E}\left[\widehat{j}_{f,n}^{K}\right] - \bar{j}_{f,n}^{K}\right| = \left|\sum_{\phi=0}^{H-1} q_n(\phi)\sum_{k=0}^{K-1}\gamma^k\left(\mathbb{E}_{(s,a)\sim\zeta_{n,\phi,k}}[f(s,a)] - \mathbb{E}_{(s,a)\sim\bar{\zeta}_{n,\phi,k}}[f(s,a)]\right)\right| \tag{28}$$

$$\leq \sum_{\phi=0}^{H-1} q_n(\phi)\sum_{k=0}^{K-1}\gamma^k\left|\mathbb{E}_{(s,a)\sim\zeta_{n,\phi,k}}[f(s,a)] - \mathbb{E}_{(s,a)\sim\bar{\zeta}_{n,\phi,k}}[f(s,a)]\right|. \tag{29}$$

For any measurable f with $|f| \leq f_{\max}$, the TV variational inequality gives

$$|\mathbb{E}_P[f] - \mathbb{E}_Q[f]| = \left|\int f\, d(P-Q)\right| \leq f_{\max}\int |dP - dQ| = 2f_{\max}\|P - Q\|_{\mathrm{TV}}. \tag{30}$$

Applying this inequality with $P = \zeta_{n,\phi,k}$ and $Q = \bar{\zeta}_{n,\phi,k}$ for every (ϕ, k) gives Eq. (8). □

A.1.3 Proof of Proposition 3.3

Proof. After selecting one of the two cost surrogates in Eq. (10), write it in the form assumed by the proposition,

$$\widehat{J}^K_{c,n} = \frac{1}{E}\sum_{e=1}^{E} X_e, \qquad 0 \le X_e \le R_n. \tag{31}$$

The variables X_e are independent across environments under the batch construction. The one-sided empirical Bernstein inequality for independent bounded variables [13] states that, with probability at least $1-\delta$,

$$\mathbb{E}\left[\frac{1}{E}\sum_{e=1}^{E} X_e\right] - \frac{1}{E}\sum_{e=1}^{E} X_e \le \sqrt{\frac{2S^2\log(2/\delta)}{E}} + \frac{7R_n\log(2/\delta)}{3(E-1)}, \tag{32}$$

where $S^2 = (E-1)^{-1}\sum_{e=1}^{E}(X_e - \widehat{J}^K_{c,n})^2$. Replacing the sample mean by $\widehat{J}^K_{c,n}$ gives Eq. (11). Thus the right-hand side of Eq. (11) is a valid β_n term for the selected surrogate. □

A.1.4 Proof of Theorem 3.4

Proof. Let $\mathcal{E}_\beta$ be the event in Proposition 3.3. On this event, Eq. (11) implies

$$\mathbb{E}\left[\widehat{J}^K_{c,n}\right] \le \widehat{J}^K_{c,n} + \beta_n. \tag{33}$$

The theorem assumes the distribution mismatch upper bound

$$J_c(\pi_n) \le \mathbb{E}\left[\widehat{J}^K_{c,n}\right] + \varepsilon_n. \tag{34}$$

Combining the two inequalities on $\mathcal{E}_\beta$,

$$J_c(\pi_n) \le \mathbb{E}\left[\widehat{J}^K_{c,n}\right] + \varepsilon_n \tag{35}$$

$$\le \widehat{J}^K_{c,n} + \beta_n + \varepsilon_n. \tag{36}$$

If the tightened surrogate constraint in Eq. (12) holds, then

$$J_c(\pi_n) \le \widehat{J}^K_{c,n} + \beta_n + \varepsilon_n \le b_c. \tag{37}$$

The event $\mathcal{E}_\beta$ has probability at least $1-\delta$, so the same probability applies to the final constraint statement. □

A.1.5 Proof of Theorem 3.5

Proof. Uniform staggered reset assigns equal phase mass $q_n(mK) = 1/M$ for $m = 0, \dots, M-1$ and zero mass to other phases. Substituting this phase distribution into the phase-mixture return from Eq. (6) gives

$$\sum_{\phi=0}^{H-1} q_n(\phi) j^K_{f,n}(\phi) = \sum_{m=0}^{M-1} q_n(mK) j^K_{f,n}(mK) \tag{38}$$

$$= \sum_{m=0}^{M-1} \frac{1}{M} j^K_{f,n}(mK) \tag{39}$$

$$= \frac{1}{M}\sum_{m=0}^{M-1} j^K_{f,n}(mK). \tag{40}$$

The final expression is the finite Cesàro mean stated in Theorem 3.5. □

A.1.6 Proof of Corollary 3.6

Proof. Lemma 3.1 gives

$$\mathbb{E}\left[\widehat{j}_{f,n}^{K}\right] = \sum_{\phi=0}^{H-1} q_n(\phi) j_{f,n}^{K}(\phi). \tag{41}$$

Theorem 3.5 evaluates the right-hand side under uniform staggered reset as the Cesàro mean. Therefore,

$$\mathbb{E}\left[\widehat{j}_{f,n}^{K}\right] = \frac{1}{M} \sum_{m=0}^{M-1} j_{f,n}^{K}(mK), \tag{42}$$

which is Eq. (13). □

A.1.7 Proof of Corollary 3.7

Proof. Start from the distribution mismatch bound in Eq. (8). Under uniform staggered reset, the only nonzero phase masses are $q_n(mK) = 1/M$ for $m = 0, \ldots, M-1$. Substituting these identities gives

$$\left|\mathbb{E}\left[\widehat{j}_{f,n}^{K}\right] - \bar{j}_{f,n}^{K}\right| \leq 2 f_{\max} \sum_{m=0}^{M-1} \frac{1}{M} \sum_{k=0}^{K-1} \gamma^k \left\|\zeta_{n,mK,k} - \bar{\zeta}_{n,mK,k}\right\|_{\mathrm{TV}} \tag{43}$$

$$= \frac{2 f_{\max}}{M} \sum_{m=0}^{M-1} \sum_{k=0}^{K-1} \gamma^k \left\|\zeta_{n,mK,k} - \bar{\zeta}_{n,mK,k}\right\|_{\mathrm{TV}}. \tag{44}$$

The final expression is Eq. (14). □

A.1.8 Proof of Corollary 3.8

Proof. Begin with the general phase-wise surrogates in Eq. (10). Uniform staggered reset gives $\phi = mK$ and $q_n(mK) = 1/M$. For every time step $t \in \{0, \ldots, MK-1\}$, there is a unique pair (m, k) with $t = mK + k$ and $0 \leq k < K$. Using the coverage mass definition in Eq. (9),

$$m_{q_n}(mK+k) = \sum_{\phi=0}^{H-1} q_n(\phi) \sum_{k'=0}^{K-1} \mathbb{I}\{mK + k = \phi + k'\} \tag{45}$$

$$= q_n(mK) \tag{46}$$

$$= \frac{1}{M}. \tag{47}$$

The coverage-correction weight in Eq. (9) is therefore

$$w_{q_n}(mK, k) = \frac{\gamma^{mK+k}}{m_{q_n}(mK+k)} = M\gamma^{mK+k}. \tag{48}$$

For the sampled cost surrogate in Eq. (10), substituting $q_n(mK) = 1/M$ and the preceding weight gives

$$\widehat{J}_{c,n}^{K,\mathrm{sample}} = \sum_{m=0}^{M-1} \frac{1}{M} \frac{1}{E_{mK}} \sum_{e:\phi_e = mK} \sum_{k=0}^{K-1} M\gamma^{mK+k} c(s_{e,mK+k}, a_{e,mK+k}) \tag{49}$$

$$= \sum_{m=0}^{M-1} \frac{1}{E_{mK}} \sum_{e:\phi_e = mK} \sum_{k=0}^{K-1} \gamma^{mK+k} c(s_{e,mK+k}, a_{e,mK+k}). \tag{50}$$

For the critic value surrogate, Eq. (10) has no coverage-correction weight. Substituting only $q_n(mK) = 1/M$ gives

$$\widehat{J}_{c,n}^{K,\mathrm{value}} = \frac{1}{M} \sum_{m=0}^{M-1} \frac{1}{E_{mK} K} \sum_{e:\phi_e = mK} \sum_{k=0}^{K-1} V_{c,n}(s_{e,mK+k}). \tag{51}$$

The two derived expressions are exactly Eq. (15). □

Table 2: Goal-task configurations. The cost limit specifies the maximum cumulative episodic cost.

Task	Agent	Horizon	Objects	Cost event	Limit
Goal	Point	1000	goal, 8 hazards, 1 vase	hazard proximity	25
Goal	Car	1000	goal, 8 hazards, 1 vase	hazard proximity	25

A.1.9 Proof of Proposition 3.9

Proof. For each phase mK, the change-of-measure identity gives

$$\bar{C}^K_{n,mK} - C^K_{n,mK} = \mathbb{E}_{s\sim d_{n,mK}}\left[(\bar{\rho}_n(s,mK)-1)j^K_{c,n,mK}(s)\right]. \tag{52}$$

Taking absolute values, applying the triangle inequality, and using $0 \le j^K_{c,n,mK}(s) \le B_c$ gives

$$\left|\bar{C}^K_{n,mK} - C^K_{n,mK}\right| \le B_c\, \mathbb{E}_{s\sim d_{n,mK}}[|\bar{\rho}_n(s,mK)-1|]. \tag{53}$$

Applying the triangle inequality to the uniform phase average and then averaging the preceding bound over $m = 0, \dots, M-1$ proves Eq. (18). □

A.1.10 Proof of Proposition 3.10

Proof. For each phase mK, change of measure gives

$$\mathbb{E}_{s\sim \bar{d}_{n,mK}}[Y_n(s,mK)] = \mathbb{E}_{s\sim d_{n,mK}}[\bar{\rho}_n(s,mK)Y_n(s,mK)]. \tag{54}$$

Therefore, the difference between the exact-ratio correction and the learned-ratio correction is

$$\mathbb{E}_{s\sim d_{n,mK}}[(\bar{\rho}_n(s,mK)-\tilde{\rho}_n(s,mK))Y_n(s,mK)]. \tag{55}$$

Taking absolute values and using $|Y_n(s,mK)| \le B_Y$ gives

$$\begin{aligned}&\left|\mathbb{E}_{s\sim \bar{d}_{n,mK}}[Y_n(s,mK)] - \mathbb{E}_{s\sim d_{n,mK}}[\tilde{\rho}_n(s,mK)Y_n(s,mK)]\right| \\ &\quad\le B_Y\, \mathbb{E}_{s\sim d_{n,mK}}[|\bar{\rho}_n(s,mK)-\tilde{\rho}_n(s,mK)|].\end{aligned} \tag{56}$$

Applying the triangle inequality to the uniform phase average and then averaging the preceding bound over $m = 0, \dots, M-1$ gives the result. □

A.2 Implementation details

We implement the Safety Gym-style Goal tasks in MJX and the Laikago locomotion task in both MJX and MuJoCo Warp to support massively parallel training on accelerators.

Goal Tasks Following Safety Gym and Safety-Gymnasium [9, 15], we implement the Point/Goal and Car/Goal tasks with MuJoCo agent models, goals, hazards, and vases. The environments are vectorized in MJX, with rewards, costs, and observations computed at each simulation step.

Both agents use two-dimensional continuous action vectors, with 12-dimensional and 24-dimensional sensor vectors for Point and Car, respectively. Observations combine these sensor vectors with three 16-bin pseudo-lidar groups for the goal, hazards, and vase, resulting in observation dimensions of 60 and 72, respectively.

Let p_t and g_t denote the agent and goal positions, respectively, and let $d_t = \|p_t - g_t\|_2$. The reward is defined as

$$r_t = d_{t-1} - d_t + \mathbb{I}\{d_t < r_{\text{goal}}\}, \tag{57}$$

where r_{goal} is the goal radius. The reward consists of progress toward the goal and a unit reward upon reaching it, after which a new goal is sampled. The binary safety cost is defined as

$$c_t = \mathbb{I}\{\min_i \|p_t - h_i\|_2 < r_{\text{hazard}}\}, \tag{58}$$

where h_i denotes the position of hazard i and r_{hazard} is the hazard radius. Both tasks use a single safety constraint with an episodic cost limit of 25.

Table 3: Algorithm configuration. Rows with shared values span all three algorithms.

Config	CPO	CRPO	TRPO-Lag
Critic learning rate		10^{-3}	
Discount factor		0.99	
Reward trace parameter		1.0	
Cost trace parameter		1.0	
Target KL		0.01	
CG iterations		15	
CG damping		0.1	
Line search coefficient		0.8	
Rollout mode		staggered	
Rollout length		$K = 10$	
Policy updates		500	
Cost limit		25	
Tightening confidence level		0.05	
Tightening ratio clip		3.0	
Hidden layers		$(64, 64)$	
Activation		tanh	
Observation normalization		true	
Lagrange optimizer	–	–	Adam
Lagrange learning rate	–	–	0.035
Initial Lagrange multiplier	–	–	0.001

Locomotion Task We adopt the Laikago locomotion task from [10] and implement it in both MJX and MuJoCo Warp. The robot tracks velocity commands on flat terrain. At the beginning of each episode, the forward and angular velocity commands are sampled as

$$v_x^{\mathrm{cmd}} \sim \mathcal{U}[-1, 2], \quad \omega_z^{\mathrm{cmd}} \sim \mathcal{U}[-0.5, 0.5], \quad v_y^{\mathrm{cmd}} = 0. \tag{59}$$

The reward penalizes squared tracking errors between commanded and actual base velocities, with an additional power regularization term.

The task uses three costs for balance, center-of-mass (CoM) height, and foot contact timing:

$$\begin{aligned} c_{\mathrm{bal},t} &= \mathbb{I}\{\theta_t > 15^\circ\}, \\ c_{\mathrm{hgt},t} &= \mathbb{I}\{h_t < 0.35\}, \\ c_{\mathrm{foot},t} &= \frac{1}{n_{\mathrm{legs}}} \sum_{i=1}^{n_{\mathrm{legs}}} \mathbb{I}\{\hat{\xi}_{t,i} \neq \xi_{t,i}\}. \end{aligned} \tag{60}$$

Here, θ_t is the angle between the robot base and world vertical axes, h_t is the CoM height, and $\hat{\xi}_{t,i}$ and $\xi_{t,i}$ denote the measured and desired contact states of leg i, respectively. The desired contact pattern follows per-leg phase variables with a fixed frequency. The foot timing constraint uses a looser limit than the balance and height constraints, as it primarily serves as a motion regularizer.

B Ablation studies

This appendix reports the Goal-task ablations that isolate the effects of staggered reset, rollout length, and constraint-tightening terms. All entries report the mean and standard deviation of the last 10 epochs over eight seeds unless otherwise specified. SC denotes the sampled cost estimate, CV denotes the critic value estimate.

B.1 Rollout length

To examine the trade-off between rollout length and environment parallelism in massively parallel safe RL, Table 4 compares increasing batch size through longer rollouts with the environment scaling results reported in Table 1.

Both approaches tend to achieve higher rewards as batch size increases, although the cost limit remains violated without tightening. Once tightening is applied, all configurations satisfy the

constraint, with increasing environment parallelism achieving higher rewards in most cases than extending rollout length. This difference is consistent with the larger number of samples collected per phase, which reduces the finite-sample bound and allows less conservative tightening. Moreover, the higher transition throughput obtained through greater parallelism reduces wall-clock time, making short-rollout training more effective in terms of both reward performance and training efficiency.

Table 4: Comparison of increasing rollout length and environment scale at the same batch size using CPO with the SC surrogate. Values are mean ± standard deviation of the last 10 updates over eight seeds. Shading indicates cost > 25. **Bold** highlights the highest feasible reward and its corresponding cost and time for each task and batch size.

Task	Batch size	Fixed envs (2k)						Fixed rollout (10)						
		Rollout	w/o tightening			w/ tightening			Envs	w/o tightening			w/ tightening		
			Reward	Cost	Time	Reward	Cost	Time		Reward	Cost	Time	Reward	Cost	Time
Point/Goal	20k	10	14.57 ±2.78	27.08 ±0.84	4.10	**7.56** ±3.06	**13.88** ±1.59	**4.09**	2k	–	–	–	–	–	–
	40k	20	17.96 ±1.12	27.05 ±1.10	7.77	**13.96** ±1.41	**15.05** ±0.63	**7.88**	4k	17.28 ±5.37	26.27 ±1.23	6.84	13.42 ±3.52	20.00 ±3.49	7.09
	80k	40	20.46 ±0.48	27.18 ±0.91	15.91	16.43 ±0.81	16.36 ±0.39	16.20	8k	21.23 ±1.05	27.02 ±1.50	12.87	**18.83** ±0.73	**21.19** ±2.05	**13.03**
Car/Goal	20k	10	25.12 ±2.11	28.63 ±1.46	12.92	**18.52** ±2.83	**14.38** ±1.18	**13.03**	2k	–	–	–	–	–	–
	40k	20	29.11 ±1.69	26.45 ±0.77	25.64	22.29 ±2.90	15.49 ±0.66	25.75	4k	27.57 ±1.76	26.87 ±1.18	24.82	**24.38** ±2.04	**18.43** ±0.75	**24.96**
	80k	40	32.55 ±0.52	26.54 ±0.74	51.22	26.76 ±1.16	16.69 ±0.70	51.32	8k	32.17 ±0.95	26.52 ±1.38	50.24	**29.57** ±0.70	**21.06** ±0.70	**50.53**

B.2 Tightening term ablation

To examine the contribution of each tightening term to constraint satisfaction, Table 5 compares CPO with the finite-sample term β and the phase-initial distribution correction term ε, applied individually and jointly to the SC and CV surrogates across environment scales from 2k to 32k.

The results show that applying the finite-sample term β accounts for most of the improvement in constraint satisfaction, whereas ε alone has little effect on cost returns. For the SC surrogate, β satisfies the constraint across all environment scales, and combining both terms provides further cost reductions in some configurations. However, the CV surrogate continues to violate the constraint in several configurations even with both terms, since critic approximation errors are not covered by the finite-sample bound. The limited effect of ε in these experiments also indicates that further evaluation on tasks with greater phase-initial distribution mismatch is needed to assess the contribution of distribution correction.

Table 5: Ablation of finite-sample tightening (β) and phase-initial distribution correction (ε) across environment scales using SC and CV surrogates. Values are mean $\pm$ standard deviation of the last 10 epochs over eight seeds. Shading indicates cost > 25.

Cost est.	Envs	No tightening		ε only		β only		Both ($\varepsilon + \beta$)	
		Reward	Cost	Reward	Cost	Reward	Cost	Reward	Cost
SC	2k	14.57 $\pm$2.78	27.08 $\pm$0.84	12.87 $\pm$3.83	25.77 $\pm$1.87	9.55 $\pm$4.08	17.43 $\pm$2.39	7.56 $\pm$3.06	13.88 $\pm$1.59
	4k	17.28 $\pm$5.37	26.27 $\pm$1.23	16.44 $\pm$3.21	27.10 $\pm$2.27	12.71 $\pm$1.99	18.19 $\pm$1.91	13.42 $\pm$3.52	20.00 $\pm$3.49
	8k	21.23 $\pm$1.05	27.02 $\pm$1.50	21.27 $\pm$0.69	27.02 $\pm$1.62	18.68 $\pm$1.43	22.69 $\pm$1.70	18.83 $\pm$0.73	21.19 $\pm$2.05
	16k	23.07 $\pm$0.46	25.73 $\pm$0.84	23.09 $\pm$0.48	25.72 $\pm$1.20	22.09 $\pm$0.55	21.86 $\pm$1.02	22.30 $\pm$0.68	22.66 $\pm$1.41
	32k	23.81 $\pm$0.64	25.07 $\pm$0.33	24.37 $\pm$0.54	25.88 $\pm$0.74	23.48 $\pm$0.70	23.19 $\pm$0.42	23.43 $\pm$0.70	23.06 $\pm$0.59
CV	2k	15.29 $\pm$3.76	29.55 $\pm$3.43	16.41 $\pm$2.95	26.62 $\pm$1.28	13.49 $\pm$2.16	24.60 $\pm$5.18	13.42 $\pm$2.06	24.53 $\pm$3.93
	4k	19.30 $\pm$2.27	29.61 $\pm$3.04	19.91 $\pm$1.09	28.82 $\pm$2.96	17.10 $\pm$1.24	23.31 $\pm$1.66	17.40 $\pm$2.23	24.57 $\pm$2.98
	8k	21.46 $\pm$1.17	27.82 $\pm$2.25	21.71 $\pm$1.58	27.16 $\pm$2.66	20.43 $\pm$2.75	26.16 $\pm$2.34	20.55 $\pm$2.29	26.91 $\pm$3.21
	16k	23.10 $\pm$0.87	26.78 $\pm$2.76	23.38 $\pm$0.59	26.07 $\pm$1.56	23.05 $\pm$0.81	25.55 $\pm$1.69	22.79 $\pm$1.18	23.94 $\pm$2.26
	32k	24.82 $\pm$0.36	26.82 $\pm$1.42	24.39 $\pm$0.73	25.37 $\pm$2.50	24.42 $\pm$0.47	25.28 $\pm$1.84	24.41 $\pm$0.86	24.92 $\pm$1.61

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes],

   Justification: The abstract and Section 1 summarize the proposed framework and its theoretical and empirical contributions.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discusses limitations in Section 5, including theoretical assumptions, reset restrictions, tightening conservatism, and critic estimation errors.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: The assumptions and theoretical statements are presented in Section 3, while complete proofs are provided in Appendix A.1.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: Section 4 and Appendix A.2 describe the task and algorithm settings, training procedures, and evaluation protocols. The GitHub repository linked in the abstract provides the code and configurations needed to reproduce the experiments.

Guidelines:

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

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

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

   Answer: [Yes]

   Justification: The code is publicly available in the GitHub repository linked in the abstract, along with setup and usage instructions.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 4 and Appendix A.2 describe the task configurations, hyperparameters, training procedures, and evaluation protocols.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Experimental results in Section 4 are averaged across multiple random seeds, and the reported error bars correspond to standard deviation across evaluation runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Section 4 specifies the hardware, and Figures 1 and 2 report wall-clock training time.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The experiments involve simulated tasks, with no human subjects or personal data.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This work may benefit safe control applications by improving training efficiency, but its contributions are methodological, with no direct societal impacts identified.

Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release high-risk models, scraped datasets, or dual-use generative systems requiring additional safeguards.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The benchmark environments and other assets used are cited, and their licenses are documented in the GitHub repository and respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: The GitHub repository linked in the abstract provides documentation, setup instructions, and experiment configurations alongside the released code.

    Guidelines:

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

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

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

    Answer:[N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

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

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: We only used LLMs for editing and formatting.

    Guidelines:

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