SAFE-HAC: MODULAR MULTI-HORIZON CRITICS FOR SAFE REINFORCEMENT LEARNING WITH OPTIONAL CONTROL ASSISTANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Safety information is heterogeneous across continuous-control tasks: some environments expose only a cost signal, others provide signed constraint margins or task-specific regulated outputs, and only some admit a conventional controller. Existing methods are typically specified around one such interface. We study safe policy learning under optional safety signals and optional controller assistance. We introduce SAFE-HAC (Soft Actor with Fallback Enhancement and Horizon-Aware Critics), a modular recurrent off-policy actor–critic whose availability mask activates any subset of three finite-horizon predictors: cumulative cost, worst future clearance, and regulated-output error. This final signal is Lyapunov-inspired only when its zero set and descent condition carry formal control-theoretic meaning. Active predictions enter the actor loss. When a fallback controller is available, the same predictions assess both proposed and fallback actions and can reject a policy proposal in favor of a backup action; without a fallback, learning remains well-defined and the policy executes directly. Because intervention changes the transition-generating action, critics train on the executed action while proposal–fallback pairs support intervention imitation. We formalize the error induced by proposal-labeled shielded replay. We evaluate both policy-only and controller-assisted modes on four continuous-control tasks against SAC-LAG, PPO-LAG, CPO, and TD3. Across these tasks, SAFE-HAC attains the highest reported mean return and lowest violation rate on `UnderwaterDrone` and `Circle2`, while the `CarGoal` environments reveal task-dependent performance–safety trade-offs alongside substantially faster convergence to safe behavior. A separate MPC-assisted study examines integrated deployment performance. Empirical evaluations demonstrate that SAFE-HAC achieves competitive task performance while reducing constraint violations and reaching the specified empirical safety criteria in fewer environment steps compared with standard constrained-RL baselines.

1 INTRODUCTION

High return is insufficient when exploration can cause collisions or violate an operating envelope. Constrained Markov decision processes (CMDPs) optimize return subject to an expected cost budget (Altman, 1999; Achiam et al., 2017). However, expected-cost constraints do not ensure safety on every trajectory, and Lagrangian updates can exhibit overshoot and oscillation (Stooke et al., 2020). An action with little immediate cost may still lead to an avoidable future violation.

A practical challenge is that *available safety resources vary across tasks*. Some environments expose only a native cost; others also provide geometric clearance or a meaningful regulation error, such as distance to a goal or deviation from a reference speed. MPC, LQR, PID, or recovery controllers may be available in some domains but absent in others. A reusable learning interface should exploit these resources without requiring all of them.

SAFE-HAC addresses this setting through modular, horizon-indexed predictors. A cost head forecasts accumulated cost, a margin head forecasts minimum clearance, and a progress head forecasts persistent regulation error. An availability mask activates the corresponding supervision, actor penalties, and optional intervention tests. With no fallback, the actor executes directly. With a

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.

fallback, the predictors screen both candidate actions before an override. Because the environment responds to the executed action, every transition is assigned to that action during critic learning; intervention imitation separately trains the actor toward the selected fallback.

Our contributions are:

- A common interface for optional cost, margin, progress, and controller resources, implemented as a recurrent actor–critic with finite-horizon auxiliary predictions.
- An intervention-aware learning rule that uses executed-action replay and fallback imitation, together with a bound on the target discrepancy induced by proposal-labeled transitions.
- A four-task policy-only evaluation and a separate controller-assisted study that distinguish standalone policy performance from integrated deployment performance.

The terms *barrier-margin* and *Lyapunov-inspired* describe prediction targets and regularizers. They do not imply verified control certificates.

2 Related Work

Constrained policy optimization. CPO uses constrained trust-region updates (Achiam et al., 2017), while FOCOPS derives first-order updates through optimization in policy space (Zhang et al., 2020). Lagrangian methods combine cost penalties with PPO or SAC (Schulman et al., 2017; Haarnoja et al., 2018; Ray et al., 2019; Ji et al., 2024); PID updates improve the response of the dual variable (Stooke et al., 2020). SAFE-HAC complements this line of work by exposing independently switchable safety signals and controller assistance. Its finite-horizon penalties are surrogates, rather than a guaranteed solver for the CMDP constraints.

Lyapunov, barrier, and predictive safety methods. Lyapunov-based RL constrains policy improvement through a Lyapunov construction (Chow et al., 2018); control barrier functions impose conditions for forward invariance (Ames et al., 2017). Learned Lyapunov and barrier actor–critics integrate related structure into policy learning (Ma et al., 2022; Du et al., 2023; Zhao et al., 2023). Safety critics, shields, and recovery policies estimate or intervene against unsafe behavior (Srinivasan et al., 2020; Alshiekh et al., 2018; Wagener et al., 2021; Thananjeyan et al., 2021; Zhao et al., 2022). Predictive safety filters use model-based backup trajectories (Wabersich and Zeilinger, 2021); RAMPS combines learned linear dynamics with robust multi-step barrier shielding (Ambadkar et al., 2026). In contrast, SAFE-HAC uses model-free recurrent forecasts and does not require a backup controller. Its expected minimum-clearance predictor is not a control barrier function or a bound on violation probability.

Control assistance and temporal prediction. Residual RL, MPC-guided learning, and action governors already combine learned and conventional control (Johannink et al., 2019; Pfrommer et al., 2022; Li et al., 2021). Our emphasis is the common interface across assisted and unassisted operation, including action-consistent replay. For partial observability, Safe SLAC uses a stochastic latent model (Hogewind et al., 2023); we instead reconstruct recurrent states from replay with burn-in (Kapturowski et al., 2019). T-SAC strengthens reward critics with sequence conditioning and multi-step returns (Tian et al., 2026), while CGPO uses finite-horizon constraint estimation for policy updates (Dai et al., 2024). Our horizon axis indexes explicitly truncated cost, clearance, and progress targets.

3 Problem Formulation

We consider a partially observed process with dynamics $x_{t+1} = F(x_t, a_t^{\text{exec}}, w_t)$ and observations $o_t = O(x_t)$. A task may provide a nonnegative native cost c_t, a signed constraint margin $m(x) = \min_j \text{clear}_j(x)$, and a regulation error

$$\mathcal{X}_{\text{safe}} = \{x : m(x) \geq 0\}, \qquad \ell_g(x) = \|g(x) - g_{\text{ref}}\|_W^2, \quad W \succeq 0. \tag{1}$$

A state- or time-dependent reference may be included in an augmented state. Depending on the task, g represents position, speed, attitude, or another regulated output. The policy observes o_t and its history; state-based labels do not imply access to the physical state at deployment.

The fixed availability mask $\boldsymbol{\chi} = (\chi_C, \chi_B, \chi_U, \chi_F) \in \{0,1\}^4$ selects cost, margin, progress, and fallback resources. It is specified before evaluation. For the recurrent information state ι_t, define

$$J_R(\pi) = \mathbb{E}_\pi \sum_{t\geq 0} \gamma^t \big[\widetilde{r}_t + \alpha \mathcal{H}(\pi(\cdot \mid \iota_t))\big], \qquad J_C(\pi) = \mathbb{E}_\pi \sum_{t\geq 0} \gamma_c^t c_t. \tag{2}$$

Here $\widetilde{r}_t$ is the training reward; reported returns use the native r_t. Any reward scaling or clipping is fixed and documented. For evaluation horizon T_e, the desired constrained problem is

$$\max_\pi J_R(\pi) \quad \text{s.t.} \quad \begin{cases} J_C(\pi) \leq d_C, & \chi_C = 1, \\ \mathbb{P}_\pi(\min_{0\leq t\leq T_e} m(x_t) < 0) \leq \delta_B, & \chi_B = 1. \end{cases} \tag{3}$$

The method uses learned surrogates for these constraints; satisfying a prediction threshold does not establish feasibility of Eq. (3). Progress alone is not a safety specification, so no safety claim is made when $\chi_C = \chi_B = 0$. Assisted evaluation concerns the composite controller, whereas policy-only evaluation concerns π itself.

4 SAFE-HAC

4.1 RECURRENT ACTOR–CRITIC

The encoder uses observations and executed-action history:

$$z_t = f_\psi\big(z_{t-1}, [o_t, a_{t-1}^{\text{exec}}, \widetilde{r}_{t-1}, b_{t-1}]\big), \qquad \iota_t = (o_t, z_t), \tag{4}$$

where b_t marks an episode boundary; the recurrent state is reset before processing the next episode's initial observation. The squashed Gaussian actor proposes $a_t^\pi \sim \pi_\theta(\cdot \mid \iota_t)$. Twin soft reward critics (Haarnoja et al., 2018) minimize

$$\mathcal{L}_Q = \mathbb{E}_{t\sim\mathcal{D}} \sum_{i=1}^{2} \big(Q_{\phi_i}(\iota_t, a_t^{\text{exec}}) - y_t^Q\big)^2, \tag{5}$$

with the standard SAC target specified in Appendix B. The bootstrap terminal flag d_t distinguishes true termination from a time-limit boundary. Only active auxiliary heads are instantiated; with all heads and the fallback disabled, the method reduces to recurrent SAC.

4.2 FINITE-HORIZON AUXILIARY PREDICTIONS

For $h \in \mathcal{H} = \{1, 4, 8\}$, let $M_{t,k} = \prod_{j=0}^{k-1}(1 - d_{t+j})$, with $M_{t,0} = 1$, and $\mathcal{K}_{t,h} = \{0 \leq k < h : M_{t,k} = 1\}$. The direct replay labels are

$$y_{t,h}^C = \sum_{k=0}^{h-1} \gamma_c^k M_{t,k} c_{t+k}, \tag{6}$$

$$y_{t,h}^B = \min_{k\in\mathcal{K}_{t,h}} m(x_{t+k+1}), \tag{7}$$

$$y_{t,h}^U = \sum_{k=0}^{h-1} \beta^k M_{t,k} \ell_g(x_{t+k+1}). \tag{8}$$

These labels include the terminal transition and exclude subsequent steps. They never cross episode boundaries. A label is used only if its required continuation is observed or ends in a true terminal state; an unavailable continuation at a time limit is not treated as zero.

Let $\mu_\mathcal{D}$ denote the historical executed-action behavior represented in replay. For each active head, define

$$X_h^{\mu_\mathcal{D}}(\iota, a) := \mathbb{E}_{\mu_\mathcal{D}}\big[y_{t,h}^X \,\big|\, \iota_t = \iota,\ a_t^{\text{exec}} = a\big], \qquad X \in \{C, B, U\}. \tag{9}$$

Here $C_h^{\mu_\mathcal{D}}$ is expected cumulative cost, $B_h^{\mu_\mathcal{D}}$ is the expected minimum successor-state margin, and $U_h^{\mu_\mathcal{D}}$ is expected accumulated regulation error over the specified horizon.

Without regularization, these quantities are the unrestricted population minimizers of the corresponding squared-error losses. The regularized predictors used below need not equal these conditional

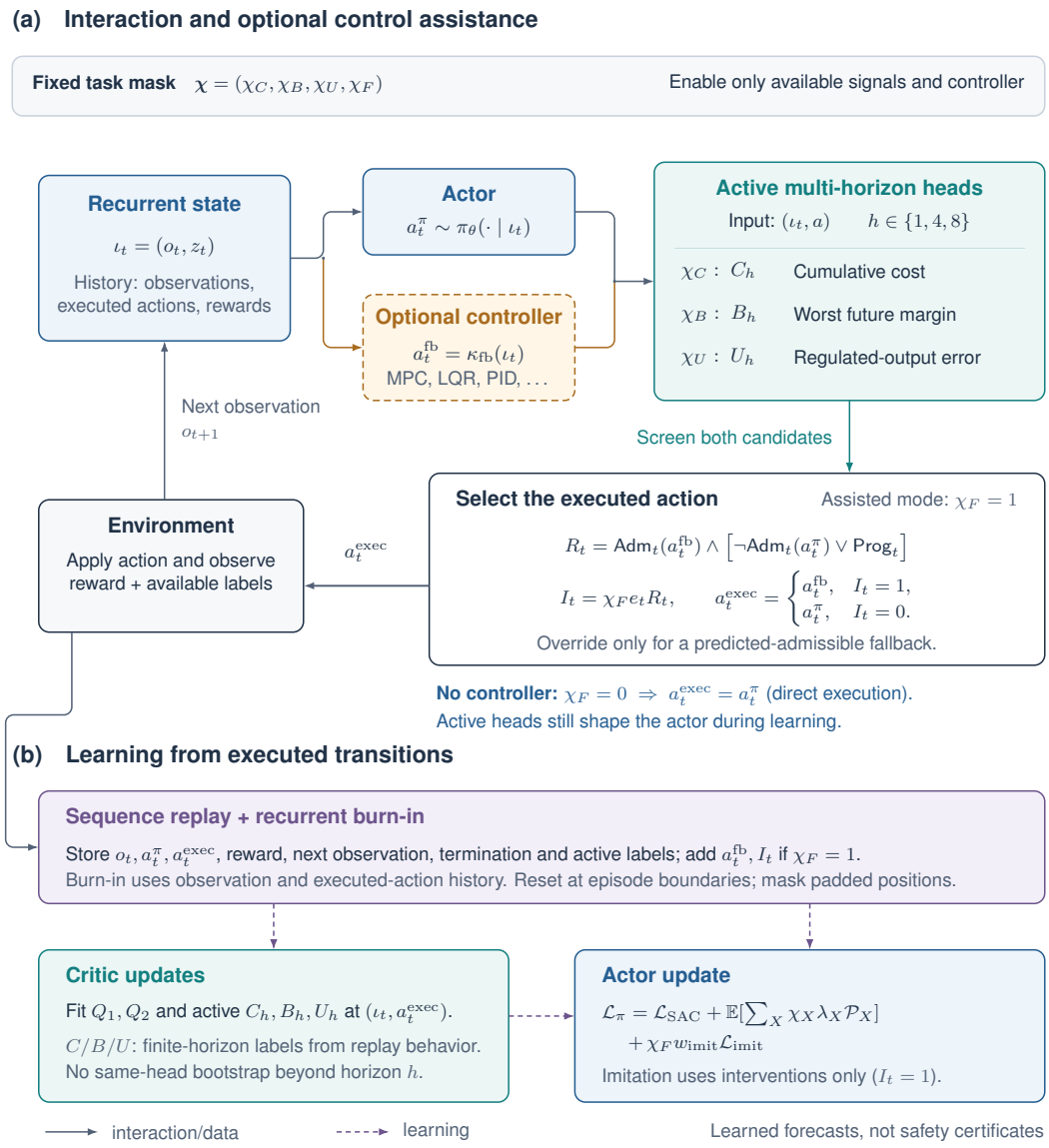


Figure 1: Availability-aware SAFE-HAC loop. Active predictors shape the policy. If $\chi_F = 1$, they also compare the proposal with an optional controller; if $\chi_F = 0$, the proposal executes directly. Replay assigns every observed transition to the executed action.

means. No same-head bootstrap is added at $t + h$, preserving the finite-horizon meaning of the labels. Their continuation follows replay behavior rather than the exact current policy; Appendix E characterizes this distinction.

Writing $e_{t,h}^X = X_h(\iota_t, a_t^{\mathrm{exec}}) - y_{t,h}^X$, we use

$$\mathcal{L}_{\mathrm{aux}} = \mathbb{E}_{t\sim\mathcal{D}} \sum_{h\in\mathcal{H}} \omega_h \sum_{X\in\{C,B,U\}} \chi_X (e_{t,h}^X)^2 + \chi_C \mathcal{R}_C + \chi_U \mathcal{R}_U, \tag{10}$$

where $\omega_h \geq 0$ and $\sum_h \omega_h = 1$. $\mathcal{R}_C$ penalizes cost underestimation; $\mathcal{R}_U$ encourages consistency with one-step regulation error and temporal descent. Their exact forms appear in Appendix C. These terms change the regression estimands: with a positive underestimation weight, the pointwise population minimizer of the asymmetric cost loss is an upper conditional expectile rather than the conditional mean (Newey and Powell, 1987). Neither a positive expected minimum margin nor a descent penalty establishes a safety or stability guarantee. Inactive terms are omitted without requesting their labels.

4.3 Actor objective and intervention imitation

For a reparameterized actor action a, collect threshold residuals as

$$(v_h^C, v_h^B, v_h^U)(\iota, a) = (C_h - \tau_h^C,\ \tau_h^B - B_h,\ U_h - \tau_h^U)(\iota, a), \qquad \mathcal{P}_X = \sum_h \omega_h [v_h^X]_+. \tag{11}$$

The thresholds specify allowed cost, minimum clearance, and allowed progress cost. They are surrogate decision thresholds, not automatic translations of d_C or δ_B. At curriculum stage s, the actor minimizes

$$\mathcal{L}_\pi^{(s)} = \mathbb{E}_{t\sim\mathcal{D},\, a\sim\pi_\theta}\Big[\alpha \log \pi_\theta(a \mid \iota_t) - \min_i Q_{\phi_i}(\iota_t, a) + \sum_{X\in\{C,B,U\}} \chi_X \lambda_X^{(s)} \mathcal{P}_X\Big] + \chi_F w_{\text{imit}} \mathcal{L}_{\text{imit}}, \tag{12}$$

$$\mathcal{L}_{\text{imit}} = \frac{\mathbb{E}_{t\sim\mathcal{D}}\big[I_t\ \text{SmoothL1}(\bar{a}_\theta(\iota_t), a_t^{\text{fb}})\big]}{\mathbb{E}_{t\sim\mathcal{D}}[I_t] + \varepsilon}, \tag{13}$$

where I_t indicates an enforced intervention and $\bar{a}_\theta$ is the squashed actor mean. Imitation is zero when no fallback is available. Penalty and enforcement schedules must be fixed before final evaluation and recorded with the settings in Appendix I.

4.4 Optional action screening

When $\chi_F = 1$, the controller supplies a_t^{fb}. It may be MPC, LQR, PID, or a recovery policy; its inputs and any privileged state access are part of the controller specification. Define $\mathcal{S} = \{X \in \{C, B\} : \chi_X = 1\}$ and predicted admissibility

$$\mathsf{Adm}_t(a) = \left[\bigwedge_{X\in\mathcal{S},\, h\in\mathcal{H}} v_h^X(\iota_t, a) \le 0\right] \wedge \neg\mathcal{R}_{\text{geom}}(\iota_t, a), \tag{14}$$

where an unavailable geometric rejection check is false. For an active progress head, write $\bar{U}_t(a) = \sum_h \omega_h U_h(\iota_t, a)$ and set $G_t = \mathbf{1}\{\bar{U}_t(a_t^\pi) > \bar{U}_t(a_t^{\text{fb}}) + \epsilon_U\}$; otherwise $G_t = 0$. An override requires a predicted-admissible fallback:

$$R_t = \mathsf{Adm}_t(a_t^{\text{fb}}) \wedge \big(\neg\mathsf{Adm}_t(a_t^\pi) \vee G_t\big), \tag{15}$$

$$I_t = \chi_F e_t R_t, \qquad a_t^{\text{exec}} = \begin{cases} a_t^{\text{fb}}, & I_t = 1, \\ a_t^\pi, & I_t = 0, \end{cases} \tag{16}$$

where $e_t \sim \text{Bernoulli}(p_{\text{shield}}(t))$ during training and $e_t = 1$ at assisted evaluation. With $\chi_F = 0$, no controller is queried and $a_t^{\text{exec}} = a_t^\pi$. If both candidates fail screening, the proposal still executes; this is an unresolved rejection, not a safe recovery. Progress-driven overrides are distinguished from safety-driven overrides. Controller details are given in Appendix F.

4.5 Action-consistent sequence replay

Replay retains observations, rewards, active labels, termination and boundary flags, and both proposed and executed actions. Assisted runs additionally store the fallback action and intervention flag. The full record is

$$\begin{aligned} &\big(o_t, a_t^\pi, a_t^{\text{exec}}, a_t^{\text{fb}}, I_t, r_t, \widetilde{r}_t, \\ &\ c_t, m(x_{t+1}), \ell_g(x_{t+1}), o_{t+1}, d_t, b_t\big), \end{aligned} \tag{17}$$

with unavailable auxiliary and fallback fields omitted. Each replay sample contains L_{burn} burn-in steps followed by L_{train} loss-bearing steps, together with the observed successor steps required by the auxiliary targets and regularizers. Recurrent states are reconstructed using executed-action histories. Burn-in is evaluated without gradient, and its final hidden state is detached before the loss-bearing unroll.

Episode boundaries reset the recurrent state. Padding, unavailable horizon targets, and unavailable continuation terms are masked separately. A target ending in a true terminal state includes the

terminal transition and excludes later steps; an unobserved continuation at a time limit is not replaced by zero. Target networks use Polyak averaging. Algorithm 1 gives the training loop.

For a fixed continuation value V, let $\mathcal{T}_V(x,a) = \mathbb{E}_w[\widetilde{r}(x,a,w) + \gamma V(F(x,a,w))]$. If reward and dynamics are action-Lipschitz with constants L_r, L_F, and V is L_V-Lipschitz, then

$$|\mathcal{T}_V(x,a^{\text{exec}}) - \mathcal{T}_V(x,a^{\pi})| \le (L_r + \gamma L_V L_F)\|a^{\text{exec}} - a^{\pi}\|. \tag{18}$$

The assumptions and proof appear in Appendix A; a recurrent continuation value requires the corresponding augmented-state assumptions. Executed-action replay removes this action-label mismatch at sampled transitions. It does not remove approximation error, replay-policy mismatch, or extrapolation error on rarely executed proposals.

5 Experiments

We examine standalone policy performance across four tasks and separately evaluate an MPC-assisted deployment on UnderwaterDrone.

Tasks and comparators. The suite contains `SafeCarGoal`, a custom dynamic CarGoal variant, partially observed `UnderwaterDrone`, and `SafetyPointCircle2-v0`, spanning static and dynamic navigation, underwater control, and orbital motion (Ray et al., 2019; Ji et al., 2023). Appendix G lists the required task specifications: observations, availability mask, labels, safety budget, and termination rules. The reported baselines are SAC-Lag, PPO-Lag, CPO, and the unconstrained TD3 reference (Fujimoto et al., 2018). FOCOPS is discussed as related work but is not included in the reported numerical comparison. Implementation sources, commits, and modifications should be recorded alongside the configurations; OmniSafe provides a common safe-RL reference (Ji et al., 2024).

Evaluation protocol. The policy-only comparison evaluates independently trained policies with $\chi_F = 0.$ seeds and 1,000 shared test scenarios per model. It compares the deployed Safe-HAC–MPC system with unshielded baselines and is analyzed separately. Within each experiment, methods share action bounds, native cost definitions, termination rules, and held-out evaluation scenarios. Interaction budgets can differ across methods, as noted in Appendix K. Training, tuning, and evaluation seeds are disjoint. Task-specific training budgets, seed counts, and evaluation episode counts must be recorded with the settings in Appendix I. Checkpoint selection prioritizes validation feasibility, then return; if no checkpoint is feasible, the minimum-cost checkpoint is retained and marked infeasible. Evaluation uses deterministic actions.

The independent training seed is the statistical unit. Table 1 reports seed means; it does not establish statistical significance. Native cost, the definition and denominator of the violation rate, and the threshold for empirical feasibility must be considered separately. In particular, a reported violation rate is not automatically the episode-level probability in Eq. (3). For seed-level intervals and paired comparisons, we specify a reporting protocol in Appendix J (Agarwal et al., 2021).

Additional margin and progress supervision is part of Safe-HAC's information budget. Comparisons with cost-only baselines therefore assess the complete method. Isolating the contribution of its optimizer or architecture requires matched-supervision and matched-capacity ablations.

6 Results

6.1 Standalone policy performance

Table 1 separates task performance from violations. On UnderwaterDrone, Safe-HAC achieves the highest mean return (411.0) and success rate (94.0%), with the lowest violation rate (5.1%). The strongest baseline return is 274.6 (PPO-Lag), while the lowest baseline violation rate is 12.9% (TD3). On Circle2, Safe-HAC also has the highest return (34.1) and lowest violation rate (1.8%); the corresponding best baseline values are 32.6 and 5.0%, both from SAC-Lag.

The CarGoal results are more mixed. On the static task, Safe-HAC reaches 99.8% success, but SAC-Lag obtains higher return and fewer violations; PPO-Lag and CPO achieve 100.0% success

Table 1: Policy-only evaluation ($\chi_F = 0$): means across training seeds. Success and violation rates are percentages. Bold denotes the best reported mean in each metric, including ties; it does not denote statistical significance or feasibility. Circle2 has no success metric. TD3 is an unconstrained reference.

	CarGoal			CarGoal-Dynamic		
Method	Return ↑	Success ↑	Viol. ↓	Return ↑	Success ↑	Viol. ↓
SAFE-HAC (ours)	488.7	99.8	5.4	592.9	**100.0**	2.2
SAC-LAG	**643.2**	98.0	2.4	648.9	**100.0**	3.2
PPO-LAG	36.9	**100.0**	0.6	509.3	**100.0**	**0.0**
CPO	36.9	**100.0**	**0.4**	**876.0**	84.0	**0.0**
TD3	630.6	91.0	1.2	627.5	99.6	9.6
	UnderwaterDrone			Circle2		
Method	Return ↑	Success ↑	Viol. ↓	Return ↑	Success	Viol. ↓
SAFE-HAC (ours)	**411.0**	**94.0**	**5.1**	**34.1**	—	**1.8**
SAC-LAG	255.9	62.0	31.2	32.6	—	5.0
PPO-LAG	274.6	71.8	28.2	27.1	—	20.2
CPO	38.2	36.2	13.7	11.6	—	6.0
TD3	75.3	21.6	12.9	26.9	—	5.2

with lower return and lower violation rates. On the dynamic task, SAFE-HAC reaches 100.0% success and 2.2% violations. CPO has the highest return and zero reported violations, but a lower success rate (84.0%); PPO-LAG combines 100.0% success with zero reported violations and lower return. These outcomes support task-dependent trade-offs rather than uniform dominance.

Crucially, final evaluation metrics mask severe differences in training-time safety and sample efficiency. Figure 2 evaluates the number of environment steps required to reach and sustain a joint condition of high task success and strict safety. On both `CarGoal` environments, SAFE-HAC converges to safe and successful behavior significantly faster than the constrained baselines. By leveraging structural multi-horizon predictions rather than relying solely on global multiplier oscillation, SAFE-HAC avoids the extended phases of unsafe exploration required by standard Lagrangian methods.

6.2 CONTROLLER-ASSISTED DEPLOYMENT

The separate Drone study evaluates SAFE-HAC with an MPC fallback active during training and deployment. Across three training seeds and 1,000 shared test scenarios per model, the assisted system attains return 412.27 ± 11.63, success $(92.67 \pm 3.40)\%$, and violation rate $(4.17 \pm 2.63)\%$ (mean $\pm$ sample standard deviation across training seeds; Table 5). In this experiment, unshielded SAC-LAG has a lower violation rate (3.73%) and a lower success rate (76.03%). These observations characterize the performance of the integrated system; they do not isolate the causal benefit of the controller or establish a safety guarantee. The associated results and training curves are provided in Appendix K.4. Figure 3 shows the corresponding training curves.

The assisted and policy-only studies report different seed counts and do not provide a matched estimate of the benefit of assistance. The relevant controls are the fallback alone, the same controller and guard paired with each policy, SAFE-HAC without imitation, and assisted-trained policies evaluated with the fallback removed. Three training seeds also provide limited evidence about between-seed variability; a large number of test scenarios does not replace independent training runs.

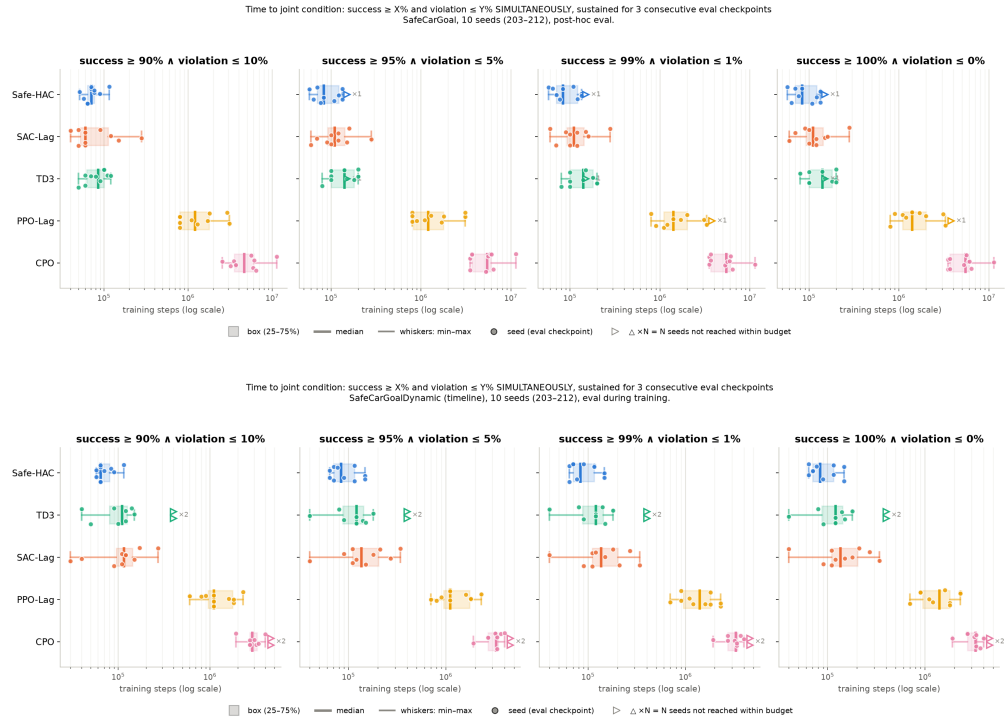


Figure 2: Training steps required to reliably reach and sustain a joint condition of high task success and low violations on `SafeCarGoal` (top) and `SafeCarGoal-Dynamic` (bottom). Note the log scale on the x-axis.

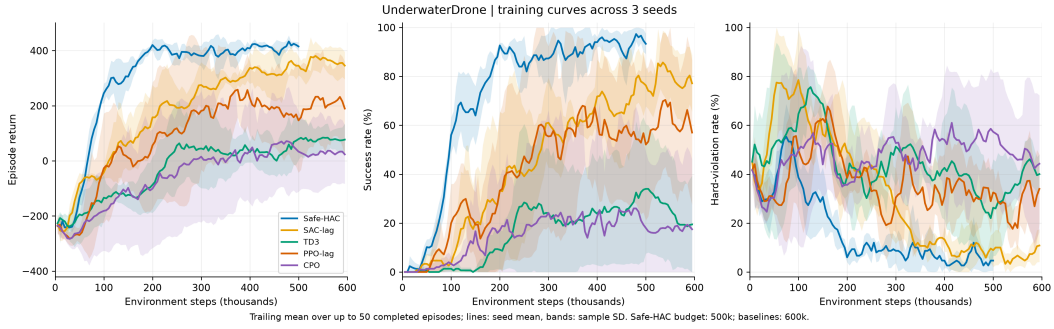


Figure 3: Training curves across the three separately repeated training seeds for `UnderwaterDrone`. Lines denote the seed mean (trailing moving average of 50 episodes), and shaded regions denote the sample standard deviation across seeds. SAFE-HAC exhibits uniquely stable task acquisition compared to all baselines.

7 LIMITATIONS

Finite-horizon forecasts can miss delayed hazards and inherit replay-policy shift, approximation error, and weak coverage of rejected actions. The learned screening rule neither certifies invariance nor guarantees the chance constraint, and it may execute an inadmissible proposal when no admissible fallback exists. Regulation error is Lyapunov-inspired only when its zero set and decrease condition have the required control-theoretic meaning. Auxiliary supervision, recurrence, and controller quality require separate ablations to attribute improvements. Finally, four simulated tasks, standalone means without numerical uncertainty intervals, and a three-seed assisted study limit conclusions about generalization and robustness.

8 Conclusion

SAFE-HAC combines optional finite-horizon cost, clearance, and progress predictors within a recurrent actor–critic. The same interface supports standalone policies and screened controller assistance, while executed-action replay preserves the correspondence between actions and observed transitions. The reported results are strongest on UnderwaterDrone and Circle2 and expose clear trade-offs on CarGoal. The method provides a modular approach to empirical safety-aware control, with certification and the isolated benefit of assistance remaining open questions.

Reproducibility Statement

Sections 3–4 specify the task interface, finite-horizon targets, actor objective, and intervention rule. Appendices B, C, and A provide optimization details, auxiliary regularizers, and the action-label discrepancy proof. Task definitions, controller specifications, and hyperparameters are referenced in Section 5. The supplementary release will include anonymous source code, frozen configurations, raw per-seed logs, checkpoint-selection code, and analysis scripts. The reporting requirements in Appendix J identify the metadata needed to reproduce and interpret the reported means.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 22–31, 2017.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29304–29320, 2021.

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Eitan Altman. *Constrained Markov Decision Processes*. Chapman and Hall/CRC, 1999.

Tanmay Sadanand Ambadkar, Darshan Chudiwal, Greg Anderson, and Abhinav Verma. Robust adaptive multi-step predictive shielding. In *International Conference on Learning Representations*, 2026.

Aaron D. Ames, Xiangru Xu, Jessy W. Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2017.

Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A Lyapunov-based approach to safe reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Juntao Dai, Yaodong Yang, Qian Zheng, and Gang Pan. Safe reinforcement learning using finite-horizon gradient-based estimation. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 9872–9903, 2024.

Desong Du, Shaohang Han, Naiming Qi, Haitham Bou Ammar, Jun Wang, and Wei Pan. Reinforcement learning for safe robot control using control Lyapunov barrier functions. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 9442–9448, 2023.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor–critic methods. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 1587–1596, 2018.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor–critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 1861–1870, 2018.

Yannick Hogewind, Thiago D. Simão, Tal Kachman, and Nils Jansen. Safe reinforcement learning from pixels using a stochastic latent representation. In *International Conference on Learning Representations*, 2023.

Jiaming Ji, Borong Zhang, Jiayi Zhou, Xuehai Pan, Weidong Huang, Ruiyang Sun, Yiran Geng, Yifan Zhong, Juntao Dai, and Yaodong Yang. Safety-Gymnasium: A unified safe reinforcement learning benchmark. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, volume 36, 2023.

Jiaming Ji, Jiayi Zhou, Borong Zhang, Juntao Dai, Xuehai Pan, Ruiyang Sun, Weidong Huang, Yiran Geng, Mickel Liu, and Yaodong Yang. OmniSafe: An infrastructure for accelerating safe reinforcement learning research. *Journal of Machine Learning Research*, 25(285):1–6, 2024.

Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 6023–6029, 2019.

Steven Kapturowski, Georg Ostrovski, John Quan, Rémi Munos, and Will Dabney. Recurrent experience replay in distributed reinforcement learning. In *International Conference on Learning Representations*, 2019.

Yutong Li, Nan Li, H. Eric Tseng, Anouck Girard, Dimitar Filev, and Ilya Kolmanovsky. Safe reinforcement learning using robust action governor. In *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, pp. 1093–1104, 2021.

Haitong Ma, Changliu Liu, Shengbo Eben Li, Sifa Zheng, and Jianyu Chen. Joint synthesis of safety certificate and safe control policy using constrained reinforcement learning. In *Proceedings of the 4th Annual Learning for Dynamics and Control Conference*, pp. 1–13, 2022.

Whitney K. Newey and James L. Powell. Asymmetric least squares estimation and testing. *Econometrica*, 55(4):819–847, 1987. doi: 10.2307/1911031.

Samuel Pfrommer, Tanmay Gautam, Alec Zhou, and Somayeh Sojoudi. Safe reinforcement learning with chance-constrained model predictive control. In *Proceedings of the 4th Annual Learning for Dynamics and Control Conference*, pp. 291–303, 2022.

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

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

Krishnan Srinivasan, Benjamin Eysenbach, Sehoon Ha, Jie Tan, and Chelsea Finn. Learning to be safe: Deep reinforcement learning with a safety critic. *arXiv preprint arXiv:2010.14603*, 2020.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by PID Lagrangian methods. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 9133–9143, 2020.

Brijen Thananjeyan, Ashwin Balakrishna, Suraj Nair, Michael Luo, Krishnan Srinivasan, Minho Hwang, Joseph E. Gonzalez, Julian Ibarz, Chelsea Finn, and Ken Goldberg. Recovery RL: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6(3):4915–4922, 2021.

Dong Tian, Onur Celik, and Gerhard Neumann. Chunking the critic: A transformer-based soft actor–critic with N-step returns. In *International Conference on Learning Representations*, 2026.

Kim P. Wabersich and Melanie N. Zeilinger. A predictive safety filter for learning-based control of constrained nonlinear dynamical systems. *Automatica*, 129:109597, 2021.

Nolan C. Wagener, Byron Boots, and Ching-An Cheng. Safe reinforcement learning using advantage-based intervention. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 10630–10640, 2021.

Yiming Zhang, Quan Vuong, and Keith W. Ross. First order constrained optimization in policy space. In *Advances in Neural Information Processing Systems*, volume 33, pp. 15338–15349, 2020.

Weiye Zhao, Tairan He, and Changliu Liu. Model-free safe control for zero-violation reinforcement learning. In *Proceedings of the 5th Conference on Robot Learning*, pp. 784–793, 2022.

Liqun Zhao, Konstantinos Gatsis, and Antonis Papachristodoulou. Stable and safe reinforcement learning via a barrier–Lyapunov actor–critic approach. In *Proceedings of the 62nd IEEE Conference on Decision and Control*, pp. 1320–1325, 2023.

A Proof of Proposition 1

Proposition 1 (Override-induced Bellman mismatch)**.** *Fix x and a bounded value function V, and let*

$$\mathcal{T}_V(x,a) = \mathbb{E}_w[\widetilde{r}(x,a,w) + \gamma V(F(x,a,w))]\,.$$

Assume uniformly in w that $\widetilde{r}$ is L_r-Lipschitz in a, F is L_F-Lipschitz in a, and V is L_V-Lipschitz in x. Terminal states have zero continuation value. If a shield replaces a^π by a^{exec}, then Eq. (18) holds. Conditioned on (x, a^{exec}), assigning the observed transition to a^{exec} instead gives an unbiased one-step sample of $\mathcal{T}_V(x, a^{\mathrm{exec}})$.

By adding and subtracting terms and applying the triangle inequality,

$$\begin{aligned}
&|\mathcal{T}_V(x,a^{\mathrm{exec}}) - \mathcal{T}_V(x,a^\pi)| \\
&\le \mathbb{E}_w\,|\widetilde{r}(x,a^{\mathrm{exec}},w) - \widetilde{r}(x,a^\pi,w)| \\
&\quad + \gamma\mathbb{E}_w\,|V(F(x,a^{\mathrm{exec}},w)) - V(F(x,a^\pi,w))| \\
&\le L_r\|a^{\mathrm{exec}} - a^\pi\| + \gamma L_V \mathbb{E}_w\,\|F(x,a^{\mathrm{exec}},w) - F(x,a^\pi,w)\| \\
&\le (L_r + \gamma L_V L_F)\|a^{\mathrm{exec}} - a^\pi\|.
\end{aligned}$$

When the replay input is a^{exec}, the observed reward and successor are generated by the same conditional environment kernel that defines $\mathcal{T}_V(x, a^{\mathrm{exec}})$; the usual one-step target is therefore unbiased conditional on (x, a^{exec}), subject to the standard assumption that the exogenous transition noise is conditionally independent of the replay-labeling operation. □

B Reward-Critic Target

For $a'_{t+1} \sim \pi_\theta(\cdot \mid \iota_{t+1})$, the twin soft reward-critic target is

$$y_t^Q = \widetilde{r}_t + (1-d_t)\gamma\Big[\min_{i\in\{1,2\}} Q_{\bar{\phi}_i}(\iota_{t+1}, a'_{t+1}) \\ - \alpha \log \pi_\theta(a'_{t+1} \mid \iota_{t+1})\Big]. \tag{19}$$

Target recurrent states are reconstructed using the same burn-in convention as online critics.

C Auxiliary Regularization Details

Using the errors defined in Section 4.2, the cost underestimation regularizer in Eq. (10) is

$$\mathcal{R}_C = \kappa_{\mathrm{under}}\,\mathbb{E}_{t\sim\mathcal{D}} \sum_{h\in\mathcal{H}} \omega_h [-e^C_{t,h}]_+^2, \qquad \kappa_{\mathrm{under}} \ge 0. \tag{20}$$

For a fixed horizon, the combined cost loss is $e^2 + \kappa_{\mathrm{under}}[-e]_+^2$. Its unrestricted conditional population minimizer is the expectile at level

$$\tau_{\mathrm{exp}} = \frac{1+\kappa_{\mathrm{under}}}{2+\kappa_{\mathrm{under}}}, \tag{21}$$

which exceeds $1/2$ when $\kappa_{\mathrm{under}} > 0$ (Newey and Powell, 1987). This asymmetric fit is neither an upper confidence bound nor a bound on violation probability.

For the progress head, the regularizer is

$$\mathcal{R}_U = \mathbb{E}_{t\sim\mathcal{D}} \sum_{h\in\mathcal{H}} \omega_h \Big(\kappa_{\mathrm{bound}}[\ell_g(x_{t+1}) - U_h(\iota_t, a_t^{\mathrm{exec}})]_+^2 \\ + \kappa_{\mathrm{desc}} M_{t,h}[U_h(\iota_{t+h}, a'_{t+h}) - \rho_h U_h(\iota_t, a_t^{\mathrm{exec}})]_+^2\Big), \tag{22}$$

where $a'_{t+h} \sim \pi_\theta(\cdot \mid \iota_{t+h})$, $0 < \rho_h \le 1$, and $\kappa_{\mathrm{bound}}, \kappa_{\mathrm{desc}} \ge 0$. All losses use valid, unpadded targets; the descent term is omitted when its continuation is unavailable or $M_{t,h} = 0$. A time-limit boundary does not supply a zero-valued continuation. The stage-bound hinge encourages consistency with the first nonnegative summand of $y^U_{t,h}$; it does not force equality with that summand. The descent term is a regularizer, not a stability theorem, and is not a bootstrap term in the finite-horizon target.

Algorithm 1 Recurrent SAFE-HAC training

1. Fix the task availability vector $\boldsymbol{\chi}$. Initialize the actor, twin reward critics, active $C/B/U$ heads, target networks, replay $\mathcal{D}$, and, only if $\chi_F = 1$, the controller interface.
2. For $t = 0, 1, \ldots, T-1$:
   - 2.1. Reconstruct ι_t and sample proposal $a_t^{\pi} \sim \pi_\theta(\cdot \mid \iota_t)$.
   - 2.2. If $\chi_F = 1$, query a_t^{fb}, evaluate both candidates with Eq. (14), compute R_t using Eq. (15), and sample the enforcement flag e_t; otherwise set $I_t = 0$.
   - 2.3. Execute a_t^{exec} using Eq. (16); observe the next transition and auxiliary labels.
   - 2.4. Store the available fields of Eq. (17) in $\mathcal{D}$.
   - 2.5. After the learning-start threshold: sample sequences, run recurrent burn-in, and construct y^Q plus targets only for active heads.
   - 2.6. Update all critics on a^{exec} using $\mathcal{L}_Q + \mathcal{L}_{\mathrm{aux}}$.
   - 2.7. Update the actor with Eq. (12), update entropy temperature if learned, and Polyak-update target networks.

Table 2: Representative configurations of the same implementation. Zero disables the corresponding resource, loss, and decision.

Mode	χ_C	χ_B	χ_U	χ_F	Interpretation
Cost only	1	0	0	0	Native-cost penalized policy
Cost + margin	1	1	0	0	Policy-only predictive safety
Full, no controller	1	1	1	0	All actor penalties; no override
Full, assisted	1	1	1	1	Predictor-gated controller assistance
Recurrent SAC	0	0	0	0	No safety claim; base architecture

D Complete Training Loop

E Characterization of Off-Policy Bias in Multi-Step Targets

The direct targets in Eqs. (6)–(8) sample the behavior-conditional quantities $C_h^{\mu_\mathcal{D}}$, $B_h^{\mu_\mathcal{D}}$, and $U_h^{\mu_\mathcal{D}}$. Here $\mu_\mathcal{D}$ includes historical actor actions and fallback interventions. Executed-action replay preserves the initial action label; the remaining off-policy difference concerns the continuation after that action.

Importance-sampling characterization. Condition on the same (ι_t, a_t) and initial conditional environment-state distribution. Assume integrable targets and that the current-policy continuation law is absolutely continuous with respect to the behavior continuation law. Define

$$\rho_{t+j} = \frac{\pi_\theta(a_{t+j} \mid \iota_{t+j})}{\mu_\mathcal{D}(a_{t+j} \mid \iota_{t+j})}, \qquad W_{t,k} = \prod_{j=1}^{k} \rho_{t+j}, \qquad W_{t,0} = 1.$$

The denominator must represent the actual conditional executed-action law. Full-history conditioning is required when ι is insufficient to specify this law. For mixed continuous and discrete action laws, ρ denotes the corresponding Radon–Nikodym derivative. After termination, use an absorbing continuation with ratio one.

Writing $\Delta C_h = C_h^{\pi_\theta} - C_h^{\mu_\mathcal{D}}$ and similarly for B_h and U_h, change of measure gives

$$\Delta C_h(\iota_t, a_t) = \mathbb{E}_{\mu_\mathcal{D}}\left[\sum_{k=1}^{h-1} \gamma_c^k M_{t,k} c_{t+k}(W_{t,k} - 1) \,\middle|\, \iota_t, a_t\right], \tag{23}$$

$$\Delta B_h(\iota_t, a_t) = \mathbb{E}_{\mu_\mathcal{D}}\left[(W_{t,h-1} - 1) \min_{k \in \mathcal{K}_{t,h}} m(x_{t+k+1}) \,\middle|\, \iota_t, a_t\right]. \tag{24}$$

The identity for ΔU_h replaces $\gamma_c^k c_{t+k}$ with $\beta^k \ell_g(x_{t+k+1})$.

These identities follow by reweighting each additive summand by its partial-path likelihood ratio, and the trajectory minimum by the complete-path ratio. Thus importance sampling can correct all three quantities when its assumptions hold; the minimum operator does not prevent correction.

Why SAFE-HAC uses unweighted targets. With $\chi_F = 1$, the behavior law is induced jointly by the historical actor, rejection rule, and fallback. For illustration, let $R(\iota)$ be the rejected proposal set and $\kappa_{\mathrm{fb}}(\iota)$ a deterministic fallback. At a fixed collection time with full enforcement ($p_{\mathrm{shield}} = 1$),

$$\mu_{\mathrm{exec}}(da \mid \iota) = \mathbf{1}_{\{a \notin R(\iota)\}} \pi_{\mathrm{beh}}(a \mid \iota)\, da + p_R(\iota)\, \delta_{\kappa_{\mathrm{fb}}(\iota)}(da),$$

where $p_R(\iota) = \Pr_{a \sim \pi_{\mathrm{beh}}(\cdot|\iota)}[a \in R(\iota)]$. The fallback therefore contributes probability mass, rather than an ordinary actor density.

Exact correction requires the historical executed-action likelihoods, including intervention probabilities and their conditional mixture in replay. Actor-density ratios alone are insufficient. Moreover, shielding can exclude continuations available to the unshielded actor, violating the support condition required for exact correction. When correction is available, products of likelihood ratios can also increase estimation variance.

SAFE-HAC therefore retains direct replay supervision: it uses observed outcomes without reconstructing these historical likelihoods and supports all three heads through the same training interface. This choice preserves behavior-conditional prediction rather than claiming exact current-policy evaluation.

Interpretation for action screening. The learned heads provide empirical forecasts for Eq. (14), with calibration assessed under the deployed closed loop and actor-only evaluation reported separately. The asymmetric cost loss encourages conservative fitting to replay labels; it is not an importance-sampling correction, and its fitted predictions need not equal the conditional means above.

F CONTROLLER INTERFACES

Controllers using the policy's information implement $a^{\mathrm{fb}} = \kappa_{\mathrm{fb}}(\iota)$; any additional inputs must be specified explicitly. For a task with a dynamics model or trusted simulator, the random-shooting MPC realization samples N sequences $A^{(j)}_{0:H-1}$, rolls them out from $\widehat{x}_t$, and returns

$$a^{\mathrm{fb}}_t = a^{(j^\star)}_0, \qquad j^\star \in \arg\min_j J_{\mathrm{MPC}}(A^{(j)}_{0:H-1}; \widehat{x}_t), \tag{25}$$

with representative cost

$$J_{\mathrm{MPC}} = \sum_{k=0}^{H-1} \Big(w_g \|p_k - p_g\|_2^2 + w_v \|v_k\|_2^2 + w_a \|a_k\|_2^2 + w_m [m_{\mathrm{safe}} - m(x_k)]_+^2 + w_{\mathrm{bad}} \mathbf{1}\{m(x_k) < 0\} \Big). \tag{26}$$

An LQR fallback reports its linearization point, (A, B, Q, R), estimator, saturation, and region of use. A PID fallback reports its regulated output, gains, anti-windup rule, derivative filtering, saturation, and reset behavior. None receives privileged state unavailable to the learned policies unless that asymmetry is explicitly reported in the controller-alone comparison.

G ENVIRONMENT AND WRAPPER DETAILS

For every environment, report the exact reward separately from all three auxiliary labels, observation normalization, action scaling, cost scaling, reset distribution, time-limit handling, and early termination. A terminal caused by a safety violation must not be conflated with an administrative truncation. For each controller, report its observation privileges, standalone return/cost, solve time, infeasibility behavior, and whether its parameters were tuned using training, validation, or test episodes.

H SAFETYPOINTCIRCLE2-V0

Let $q_t = (x_t, y_t)$, $\boldsymbol{v}_t = (u_t, v_t)$, and $\varrho_t = \sqrt{x_t^2 + y_t^2}$. In the evaluated environment version, the native per-step reward is

$$r^{\mathrm{circle}}_t = 0.1 \frac{-u_t y_t + v_t x_t}{\varrho_t \big(1 + |\varrho_t - 1.5|\big)}. \tag{27}$$

Table 3: Task interfaces used in the completed Circle2, UnderwaterDrone, and Car experiments. The mask is ordered as $(\chi_C, \chi_B, \chi_U, \chi_F)$ and describes the enabled Safe-HAC components. Training steps are reported per seed.

Environment	Package/version	Horizon	χ	Cost/margin	$g(x)$	Fallback	Budget/steps
SafeCarGoal	`Calf_Barrier_Safe_HAC_Code` snapshot[d]	200	$(1,1,1,0)$	Indicator cost $\mathbf{1}[m<0]$; $m(q)=\min(d_{\text{hazard}}-0.70, 5.0-\max\{\lvert x\rvert,\lvert y\rvert\})$	Target distance	None	Method-specific safety settings; 147.2k Safe-HAC / 300k–13.1M baselines
SafeCarGoal-Dynamic	Same snapshot (dynamic variant)[d]	200	$(1,1,1,0)$	Same as SafeCarGoal; moving hazards (amplitude 0.42, period 180)	Target distance	None	Method-specific safety settings; 179.2k Safe-HAC / 250k–5M baselines
UnderwaterDrone	Custom; frozen source snapshot[a]	1500	$(0,1,1,1)$	Dense proximity cost plus obstacle-entry indicator; minimum signed obstacle clearance	Target-relative position; $\ell_g=\lVert q-q_{\text{goal}}\rVert_2^2$	MPC with learned gate	Method-specific safety settings; 500k Safe-HAC / 600k baselines
SafetyPoint Circle2-v0	Safety-Gymnasium + custom wrapper[b]	500	$(1,1,1,0)$	Boundary-crossing cost; $m(q)=1.125-\max\{\lvert x\rvert,\lvert y\rvert\}$	Reference-superellipse position and tangential-velocity errors[c]	None	Method-specific safety settings; 100k Safe-HAC / 200k baselines

[a]The UnderwaterDrone source snapshot is distributed under `reproduction/underwater_final/`, with file hashes in `SOURCE_MANIFEST.json`. The environment supplies a cost signal, but the evaluated Safe-HAC configuration disables the cost critic ($\chi_C=0$). Its progress label does not include velocity or attitude.

[b]Circle2 training provenance corresponds to repository commit `df0be561bfab5984920c71b280bb489b3e5a6166` with local working tree modifications. The exact wrapper and environment snapshot used during training are preserved in the supplementary code archive.

[c]The reference superellipse has exponent $p=3$ and radius $r_{\text{ref}}=1.025$. The counter-clockwise reference tangent velocity has magnitude 1.5. The progress label is:

$$\ell_g(q,v)=\left(\frac{(|x|^3+|y|^3)^{1/3}-1.025}{1.025}\right)^2+\frac{\|v-v_{\text{ref}}(q)\|_2^2}{1.5^2}$$

Safety budgets are method-specific: for example, Circle2 PPO-Lag uses an episodic cost limit of 0, whereas CPO uses 0.1. Safe-HAC critic thresholds are not interchangeable with episodic cost budgets.

[d]Car environments were executed using the `Calf_Barrier_Safe_MHLB_Code` codebase (branch `aida/resume-0916`, base commit `53dfb16`) with local modifications including the Random-Shooting MPC shield port. The main reported runs evaluate the learned policy without the fallback shield active ($\chi_F=0$).

Experimental protocol note. This table summarizes completed evaluation runs. Because training budgets and fallback controller settings (χ_F) differ across methods and domains (e.g., Safe-HAC utilizes an MPC fallback in UnderwaterDrone), these results report environment-specific configurations rather than a strictly matched-budget, policy-only comparison.

The signed numerator divided by ϱ_t is counter-clockwise tangential speed; the second denominator factor penalizes radial deviation from the native radius 1.5. Native safety cost is separate. The admissible region is the square $|x|\le 1.125$, $|y|\le 1.125$, giving margin

$$m(q)=\min\{1.125-|x|,\ 1.125-|y|\}. \tag{28}$$

On boundary crossing, the evaluation wrapper sets cost to one, replaces the terminal reward by -1, and terminates the episode.

The evaluated SAFE-HAC configuration uses a safe reference superellipse

$$r_p(q)=\left(|x|^p+|y|^p\right)^{1/p}=r_{\text{ref}},\qquad p=3,\quad r_{\text{ref}}=1.025. \tag{29}$$

Let $\boldsymbol{v}_{\text{ref}}(q)$ be the counter-clockwise unit tangent to the level set, scaled to magnitude 1.5. The nonnegative progress label is

$$\ell_g(q,\boldsymbol{v})=\left(\frac{|r_p(q)-r_{\text{ref}}|}{r_{\text{ref}}}\right)^2+\left(\frac{\|\boldsymbol{v}-\boldsymbol{v}_{\text{ref}}(q)\|_2}{1.5}\right)^2. \tag{30}$$

The recurrent progress critic uses $\mathcal{H}=\{1,4,8\}$ and $\beta=0.99$. In the current configuration, the stage-bound and descent regularizers have weights 0.01 and 0.1, respectively, with descent factor $\rho_h=0.99^h$. After 5,000 steps, the actor receives a progress penalty of weight 0.1 above threshold $U=2$.

I Architectures and Hyperparameters

Table 4: Selected experimental settings and SAFE-HAC hyperparameters. Training budgets are configured environment-transition counts. Complete method-specific configurations, architecture details, reward definitions, preset overrides, seed lists, and evaluation protocols are provided in the anonymous supplementary material, together with realized run lengths and any truncation of plotted training curves.

Setting	CarGoal	CarGoal Dynamic	Circle2	Drone (assisted)
Environment and training budgets (timesteps)				
Training episode limit	200	200	500	1,500
SAFE-HAC budget	150k	180k	100k	500k
SAC-LAG budget	300k	500k	200k	600k
TD3 variant budget	300k	500k	200k[†]	600k
PPO-LAG budget	10M	3M	200k	600k
CPO budget	13.12M	13.12M	200k	600k
SAFE-HAC architecture and optimization				
MLP hidden widths	$(400, 400)$	$(400, 400)$	$(512, 640, 256)$	$(512, 640, 256)$
Recurrent cell / size	GRU / 256	GRU / 256	GRU / 256	GRU / 256
Prediction horizons $\mathcal{H}$	$\{1, 4, 8\}$	$\{1, 4, 8\}$	$\{1, 4, 8\}$	$\{1, 4, 8\}$
Discounts $(\gamma, \gamma_c, \beta)$	$(0.99,\ 0.99,\ 0.99)$			
Actor learning rate	10^{-3}	$\approx 6{\times}10^{-4}$	$3{\times}10^{-4}$	$2{\times}10^{-4}$
Critic learning rate	10^{-3}	$\approx 6{\times}10^{-4}$	10^{-3}	$3{\times}10^{-4}$
Batch size	128	128	128	128
Replay capacity	300k	300k	100k	310k
Burn-in / training length	8/24	8/24	8/32	8/32
Learning-start threshold	5k	5k	6k	6k
Target update coefficient	0.005	0.005	0.005	0.005
Reward scaling ρ_r	0.05	0.05	1.0	0.05
Absolute reward clip	50	50	50	50
External fallback	None	None	None	MPC

Notes. k and M denote 10^3 and 10^6 transitions, respectively. CarGoal and CarGoal Dynamic denote the static and dynamic `SafeCarGoal` tasks; Circle2 denotes `SafetyPointCircle2-v0`. The dynamic CarGoal actor and critic learning rates are both $5.995921198{\times}10^{-4}$ before rounding.
Circle2 uses five training seeds and actor-only held-out evaluation. The assisted Drone study uses three training seeds; both studies evaluate 1,000 held-out episodes per trained model. Drone evaluates the sampled SAFE-HAC actor with shielding and MPC, whereas its baselines use deterministic inference. The MPC planning horizon is 12, with 1,000 candidate action sequences.
[†]The Circle2 TD3 reference uses a fixed cost penalty of 10 and is therefore denoted TD3-PENALTY.

J Reporting and Statistical Details

For each task and method, report the training seeds, interaction budget, selected checkpoints, number of evaluation episodes, and availability mask. Define native cost, success, and the violation-rate numerator and denominator separately. Compute evaluation metrics within each trained model before aggregating across independent training seeds. The assisted results in Table 5 use the mean and sample standard deviation of these seed-level metrics.

Seed-level confidence intervals should resample independent training runs; paired comparisons require a documented pairing across methods. Shared evaluation scenarios alone do not make independent training runs paired. Scenario-level Wilson intervals quantify uncertainty for a fixed evaluated policy and do not capture training-seed variability. The main table reports means only, so it does not support claims of statistical significance (Agarwal et al., 2021).

For time-to-feasibility comparisons, prespecify the safety and success thresholds, evaluation cadence, and required number of consecutive qualifying checkpoints. Report the first checkpoint satisfying

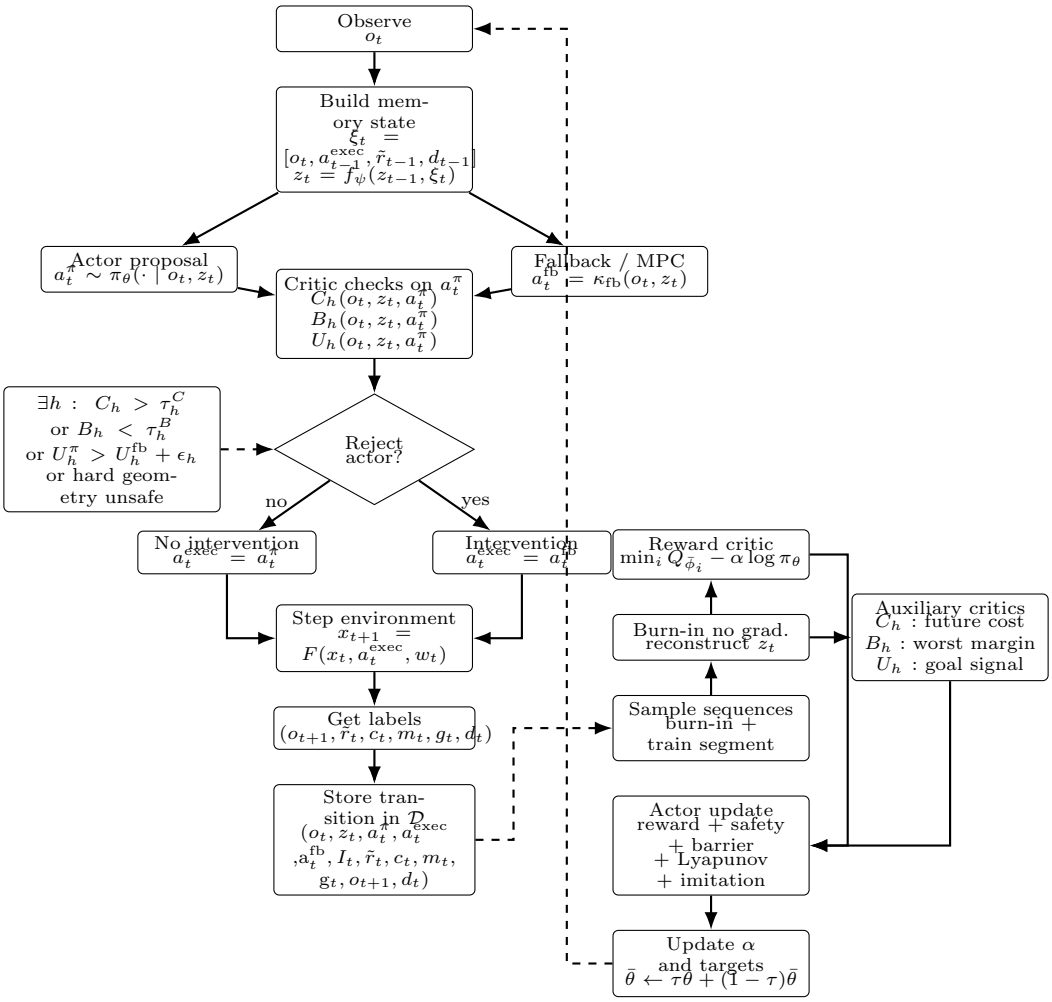


Figure 4: Closed-loop data flow of the recurrent safe off-policy actor–critic. The actor proposes a_t^π, the shield may replace it with a_t^{fb}, and replay stores the executed action a_t^{exec}, which is the action that actually generated the transition.

this sustained criterion. Runs that never satisfy it are right-censored at their training budgets; retain all such runs and report the censored fraction. Cumulative native training cost should be reported separately, with interaction budgets stated explicitly. These are reporting requirements; missing configuration values and uncertainty estimates must be recovered from the original logs.

K Additional Training and Evaluation Results

This section presents training dynamics, variability, and sample-efficiency comparisons across the reported seeds.

K.1 SafeCarGoal

Figure 5 presents the per-algorithm learning curves on `SafeCarGoal`, detailing the median, interquartile range (IQR), and min–max bounds for both stochastic training rollouts and deterministic evaluation. Because total environment interactions vary drastically between off-policy and on-policy methods, Figure 6 superimposes the smoothed median evaluation metrics on a logarithmic x-axis. These curves compare task learning and violation reduction over training; final performance and safety trade-offs are reported in Table 1.

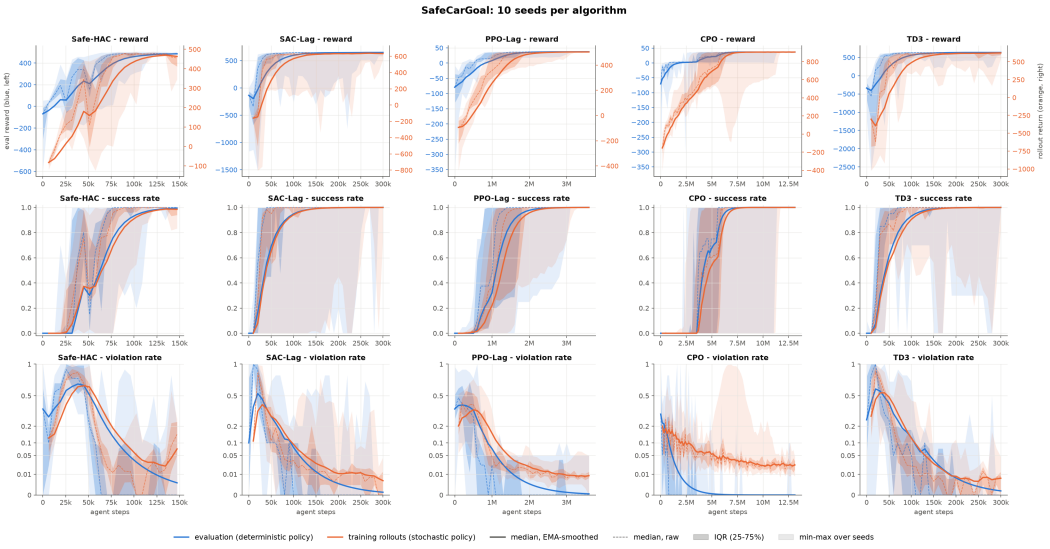


Figure 5: Detailed training and evaluation dynamics for `SafeCarGoal` across 10 independent seeds. Note the differing x-axis scales, as on-policy methods required millions of steps to converge.

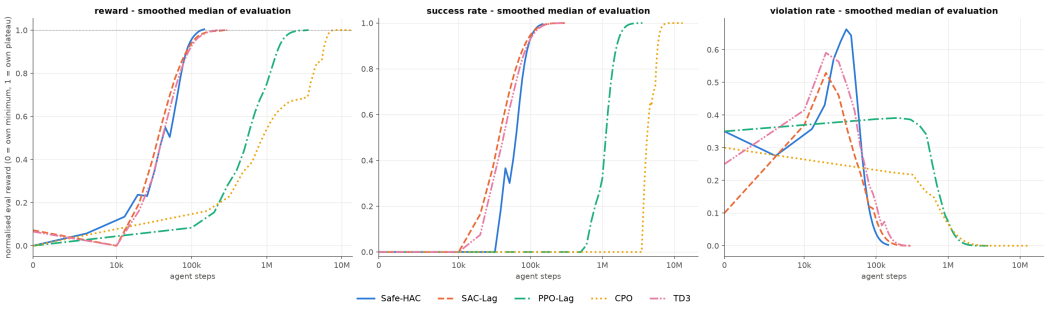


Figure 6: Smoothed median evaluation metrics for `SafeCarGoal` (logarithmic x-axis). The panels compare return, task success, and violation rate during training.

K.2 SafeCarGoal-Dynamic

Figures 7 and 8 present the corresponding training dynamics for the `SafeCarGoal-Dynamic` environment. Dynamic obstacles make safe navigation more demanding. The curves show how return, success, and violation rates evolve during training; final task-dependent trade-offs are summarized in Table 1.

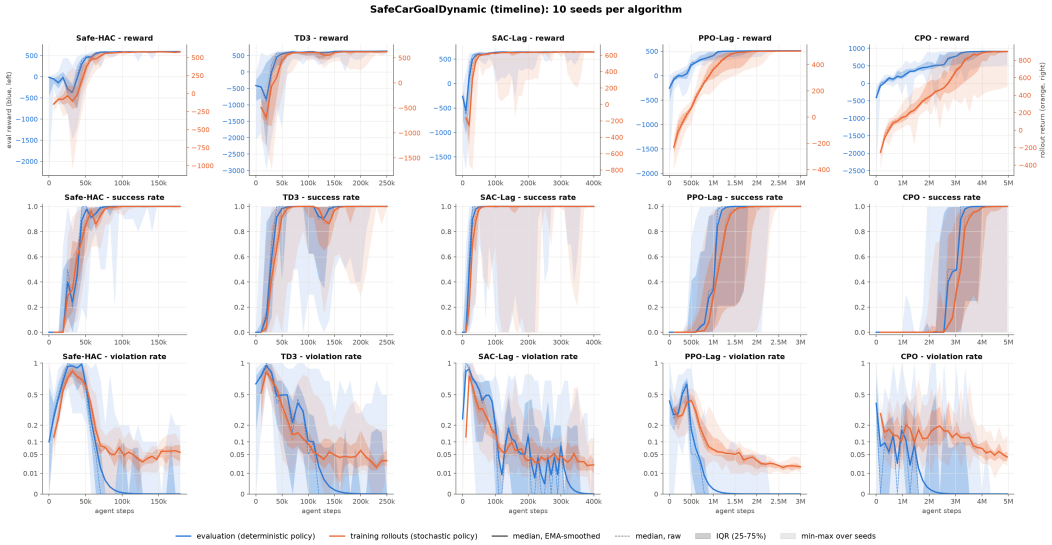


Figure 7: Detailed training and evaluation dynamics for `SafeCarGoal-Dynamic` across 10 independent seeds.

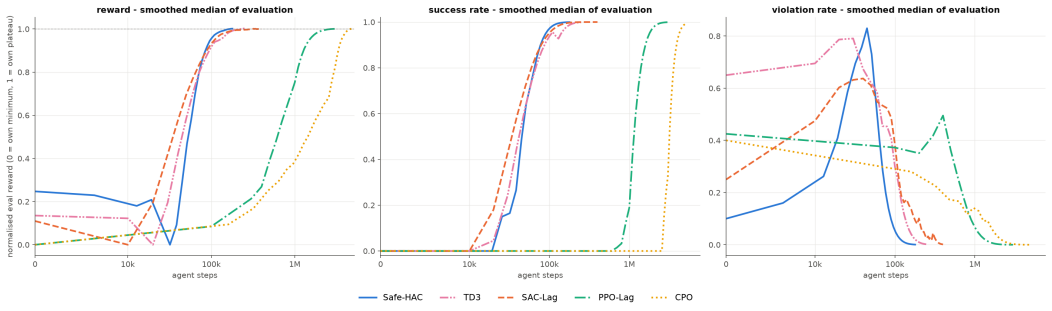


Figure 8: Smoothed median evaluation metrics for `SafeCarGoal-Dynamic` (logarithmic x-axis).

K.3 UnderwaterDrone: Policy-Only Evaluation

The `UnderwaterDrone` task introduces complex continuous control dynamics where safety violations are historically difficult to avoid during early exploration. Figure 9 aggregates both the training dynamics and final test-time safety. The training panel shows an illustrative run (seed 203). The evaluation uses 1,000 shared scenarios per model; Safe-HAC has the lowest reported mean violation rate (5.1%), as summarized in Table 1. This comparison does not establish statistical significance across training seeds.

To visualize the learned policies, Figure 10 plots the spatial trajectory distributions across these 1,000 evaluation episodes. The visitation heatmaps illustrate navigation around the hazardous region (grey) toward the goal (green). They provide qualitative context for the success and violation rates in Table 1; trajectory density alone does not establish safety.

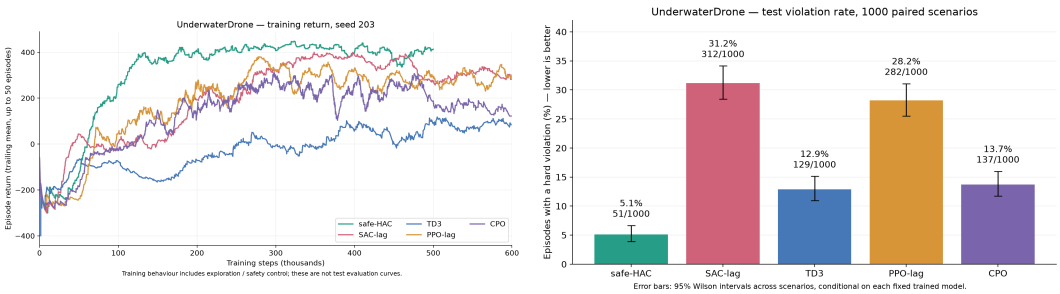


Figure 9: Left: Illustrative training return dynamics (seed 203) on `UnderwaterDrone`. Right: Test violation rate across 1,000 paired evaluation scenarios (error bars denote 95% Wilson intervals). The Wilson intervals describe scenario-level uncertainty and do not measure variability across independent training seeds.

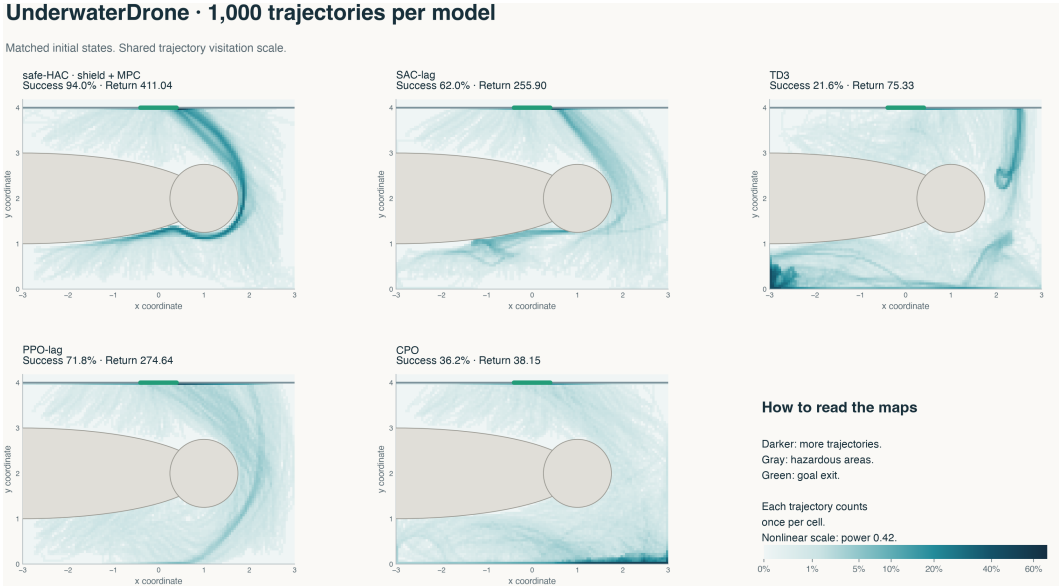


Figure 10: Spatial trajectory visitation distributions over 1,000 evaluation episodes with matched initial states. Darker blue indicates higher visitation frequency; the grey areas represent environmental hazards, and the green segment marks the goal exit. These distributions illustrate differences in path concentration and goal-directed behavior.

K.4 UnderwaterDrone: Controller-Assisted Evaluation

In this experiment, Safe-HAC is trained and evaluated with an active MPC fallback ($\chi_F = 1$), whereas the baselines are evaluated without controller assistance. Table 5 reports results across three independent training seeds (203, 307, 503), with 1,000 shared test scenarios per trained model.

The assisted Safe-HAC system achieves the highest reported mean return and success rate in this experiment. Its sample standard deviations are smaller than those of the evaluated baselines for all three reported metrics. SAC-Lag achieves the lowest mean violation rate, 3.73%, compared with 4.17% for assisted Safe-HAC.

These results characterize the integrated policy–controller system on the evaluated training seeds. They do not isolate the contribution of MPC, because the comparison does not hold controller assistance fixed across methods. The observed variability across three training seeds also does not establish robustness across other initializations.

Table 5: Controller-assisted evaluation on `UnderwaterDrone` across three training seeds (1,000 test scenarios per model). Metrics are reported as Mean ± Sample SD across the training seeds.

Method	Return	Success (%)	Violations (%)
SAFE-HAC(Shielded)	$\mathbf{412.27 \pm 11.63}$	$\mathbf{92.67 \pm 3.40}$	4.17 ± 2.63
SAC-LAG	354.60 ± 63.89	76.03 ± 17.87	$\mathbf{3.73 \pm 3.16}$
TD3	55.61 ± 22.95	16.97 ± 14.90	24.63 ± 12.23
PPO-LAG	188.40 ± 194.42	61.07 ± 29.31	22.77 ± 6.27
CPO	2.74 ± 44.36	12.23 ± 20.76	45.43 ± 41.95

K.5 SAFETYPOINTCIRCLE2-V0

Figures 11 and 12 present the aggregated training and evaluation results for the `Circle2` task across five independent seeds. The reported training budget is 100k steps for SAFE-HAC and 200k steps for each baseline. Figure 11, panel B, compares hard-violation rates, while Figure 12 compares held-out returns. These figures summarize five training seeds; their correspondence to the aggregate in Table 1 must be verified from the original logs.

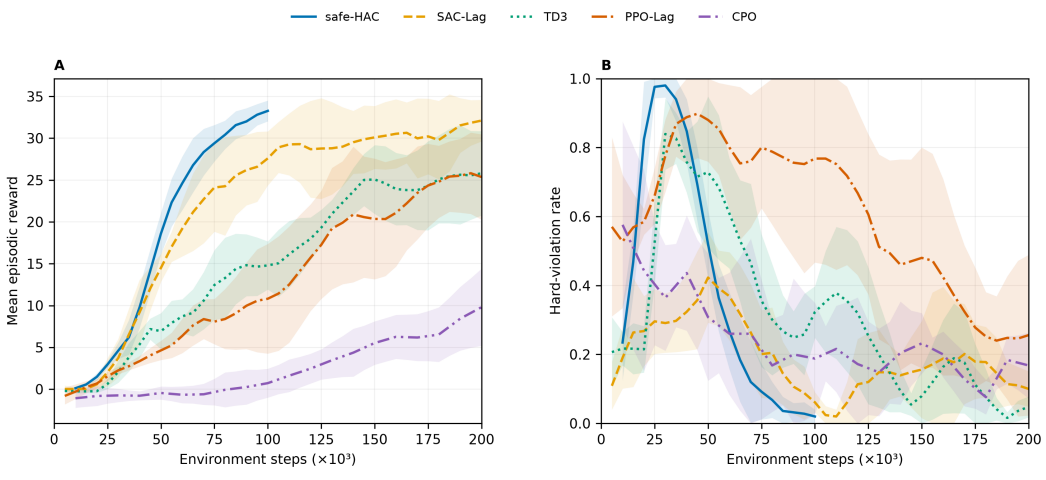


Figure 11: Five-seed aggregated training curves for `Circle2`. Panel A shows mean episodic reward, and Panel B shows hard-violation rates. Training budgets are 100k steps for SAFE-HAC and 200k steps for the baselines.

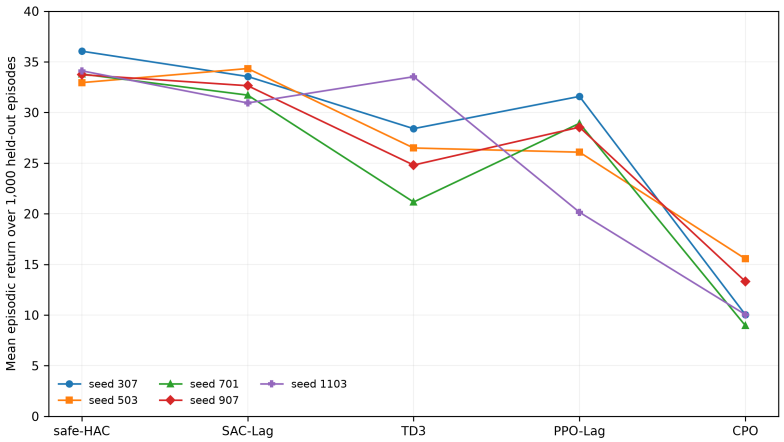


Figure 12: Held-out episodic return on 1,000 common reset seeds for `Circle2`. The figure compares deterministic evaluation policies across five training seeds.