ONLINE CONFORMAL SHIELDING: SAFE ADAPTATION UNDER COUPLED DISTRIBUTION SHIFTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Safe online adaptation requires balancing useful exploration with execution-time risk control. We identify two challenges: a *safety–exploration imbalance* between overly conservative and overly permissive control, and *safety staleness*, where risk control becomes outdated under coupled exogenous environment shifts and endogenous policy/scorer updates. We propose *Online Conformal Shielding (OCS)*, a pre-execution safety framework for online adaptation. OCS selectively executes proposed actions or intervenes, updating its threshold from partial safety feedback while the task policy and scorer continue to adapt. Our *generic reduction to online conformal abstention* (OCA) transfers suitable selective-risk guarantees to shielding without explicitly modeling the source of distribution shift. We further develop a *prior-aware OCA variant* that uses pre-adaptation information to guide early threshold selection, with a well-placed prior improving the resulting safety guarantee. Under non-anticipating interaction, we establish a high-probability *adaptive safety guarantee*: at a prescribed interaction budget, the realized average safety cost among executed task actions is bounded by the target level plus an explicit finite-sample slack. On VLA manipulation in ManiSkill and continuous control in Safety-Gymnasium, OCS keeps the selective safety risk near the target level under coupled shifts, where policy-level safe RL and fixed shields exceed it, while executing over 96% of the steps on ManiSkill.

1 INTRODUCTION

Adapting a pretrained policy to a new environment requires interaction: the agent must collect experience where its existing knowledge is incomplete. This is particularly important for vision-language-action (VLA) policies in robotics, where changes in visual appearance, physical environments, or robot dynamics can undermine performance after deployment. Online reinforcement learning offers a practical route to improvement, but the exploration that supply adaptation data may themselves cause collisions, object damage, or other costly failures. *Safe adaptation* therefore demands more than a safe final policy: it must control execution-time risk while preserving enough interaction for learning.

A first challenge is the *safety–exploration imbalance*. Policy-level safe RL shapes or constrains the task policy to limit safety costs (Achiam et al., 2017; Ray et al., 2019). Tight constraints can suppress useful exploration, whereas loose constraints can permit unsafe executions before observed costs correct the policy. Runtime shielding mitigates this tension by separating action proposal from execution: the task policy continues to propose exploratory actions, while a shield selectively intercepts proposals judged too risky (Alshiekh et al., 2018). This separation enables immediate intervention without waiting for the task policy to become safe, but its effectiveness depends on whether the shield remains reliable. This introduces a second challenge, *safety staleness*: a previously valid safety mechanism may no longer provide its intended risk control as the interaction process changes. Deployment introduces *exogenous shifts*, such as new visual conditions or transition dynamics. Adaptation then induces *endogenous shifts*, because policy and safety-risk-scorer updates change both the visited state–action distribution and the scores used for shielding. We refer to their joint effect as *coupled distribution shifts*. A fixed calibration threshold can become stale, while model-based or certificate-based shields remain valid only as long as their supporting assumptions hold (Alshiekh et al., 2018; Bastani, 2021; Banerjee et al., 2024). Safe adaptation thus requires not only selective execution, but also safety control that remains valid as the closed-loop system evolves.

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

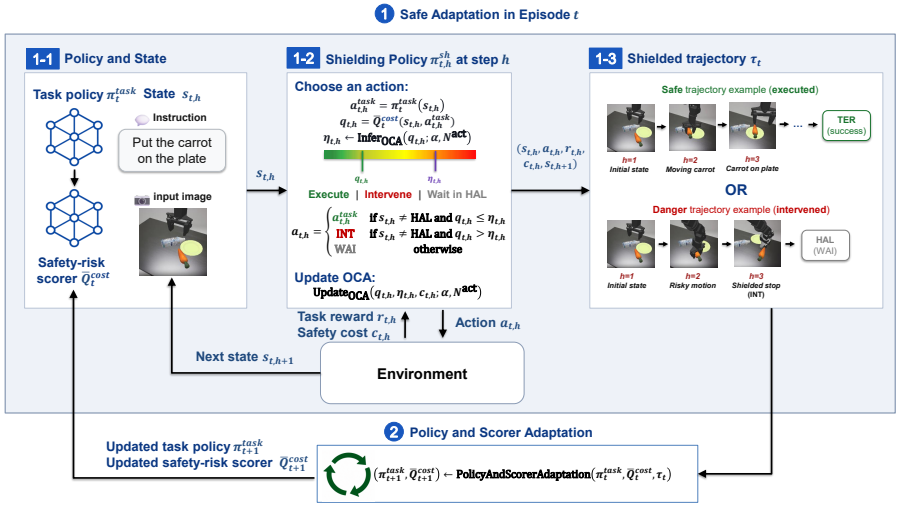


Figure 1: Overview of OCS. At each step, the task policy proposes an action, the safety-risk scorer scores it, and an OCA-sampled threshold decides whether to execute it or intervene (INT). The shielded trajectory then updates the task policy and the scorer.

We propose *Online Conformal Shielding* (OCS), which adapts an execution threshold online while the task policy and safety-risk scorer continue to learn from shielded interaction. The shield either executes a proposed action or intervenes, observing the proposed action's safety cost only when it is executed. Figure 1 illustrates the OCS framework. Our framework provides a *generic reduction from OCS to online conformal abstention* (OCA) (Lee et al., 2026), transferring suitable partial-feedback guarantees to execution-level risk control under coupled shifts. This reduction accommodates different policy and scorer update rules without requiring an accurate dynamics model or a safe initial task policy. We further develop a *prior-aware OCA variant for fast safe adaptation*: information available before interaction guides early threshold selection, with a well-placed prior improving the resulting safety guarantee. Finally, we establish a high-probability *adaptive safety guarantee*: at a prescribed interaction budget, the realized average safety cost among executed task actions is bounded by the target level plus an explicit finite-sample slack. Under non-anticipating interaction, this slack vanishes as the budget grows when the executed-task fraction stays bounded away from zero.

We evaluate OCS on VLA manipulation in ManiSkill and continuous control in Safety-Gymnasium, using constrained-RL updates for policy adaptation. Under deployment and policy-induced shifts, OCS keeps the selective safety risk within 0.018–0.029 for a target level of 0.025 across five ManiSkill environments, while executing 97% of the proposed actions and reaching 66% task success on average. Compared with PPO-Lag, a standard policy-level safe RL method, it reduces the selective safety risk by 44% and improves task success by 11 percentage points, whereas conformal policy control exceeds the target in every environment. These results demonstrate how online shielding balances execution-time risk control with useful interaction for continued adaptation.

2 Related Work

We introduce closely related work on safe adaptation. Appendix A discusses conformal safety control and compares the safety assumptions and shift robustness of representative methods (Table 1).

Policy-level Safe RL. Constrained RL, including CPO (Achiam et al., 2017) and PPO-Lag (Ray et al., 2019), optimizes task return subject to expected cumulative safety-cost constraints. Provably safe exploration methods extend safety guarantees to learning (Ding et al., 2020; Bura et al., 2022; Xiong et al., 2023; Wagener et al., 2021), under method-specific assumptions (Table 1). OPDOP assumes linear transition, reward, and utility models; DOPE and SUCBVI require finite state–action spaces, with DOPE also requiring a safe baseline. SAILR requires a backup policy and a safety critic admissible under the true dynamics. VLA post-training also uses online RL (Liu et al., 2025; Lu et al., 2025; Chen et al., 2025), including safety-constrained adaptation (Zhang et al., 2025). Policy-only methods can face the *safety–exploration imbalance*: tight constraints suppress useful exploration,

while loose ones permit unsafe execution until feedback corrects the policy. Their guarantees need not survive dynamics changes outside their assumptions. OCS instead separates exploratory action proposals from execution decisions, controlling realized safety risk among executed task actions for general bounded costs without finite/linear environment assumptions or a safe initial task policy.

Runtime Shielding and VLA Safety. Shielding monitors a policy at runtime and replaces a proposed action when it may violate a safety specification (Alshiekh et al., 2018). Such selective intervention can mitigate the *safety–exploration imbalance* by retaining admissible exploratory actions. Classical and model-predictive shields rely on sound finite abstractions or dynamics models with invariant-set and recoverability conditions (Bastani, 2021; Li & Bastani, 2020), while safe exploration with policy priors requires a conservative prior policy and a calibrated dynamics model (McDonald et al., 2026). MPS supports continuous systems, and DMPS (Banerjee et al., 2024) adds planned recovery, but both retain model and backup requirements. Such certificates can suffer from *safety staleness* when exogenous environment changes invalidate their assumptions. Adaptive shields that update their safety model (Kwon et al., 2025) mitigate this only under structural assumptions on the dynamics and cost function. VLA safety methods detect failures or correct actions at inference time (Gu et al., 2025; Hu et al., 2025; Yang et al., 2025); SAFE (Gu et al., 2025) uses conformal failure detection for a fixed policy. These results do not establish risk control throughout online policy adaptation. OCS retains intervention but requires no calibrated dynamics model or certified safety critic. Under Assumption 1, its online updates control selective risk under coupled distribution shifts (Theorem 1).

3 Problem: Safe Adaptation under Coupled Distribution Shifts

We consider safe reinforcement learning with intervention for online adaptation to a new target environment. Deployment from the pretraining environment to the target environment introduces an exogenous distribution shift. During adaptation, a target MDP remains fixed, while updates to a policy induce endogenous shifts in the interaction distribution. To this end, we view the target environment as a constrained Markov decision process $\mathcal{M} \coloneqq (\mathcal{S}, \mathcal{A}, p, p_1, \mathcal{R}, \mathcal{C}, \alpha, \gamma_R, \gamma_C, H)$, where $\Delta(\mathcal{X})$ denotes the set of probability distributions over $\mathcal{X}$, $\mathcal{S}$ is a state space, $\mathcal{A}$ is an action space, $\mathcal{R}$ is a set of reward values, $\mathcal{C} \coloneqq [0,1]$ is a set of safety cost values, $p : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S} \times \mathcal{R} \times \mathcal{C})$ is a transition kernel, $p_1 \in \Delta(\mathcal{S})$ is an initial state distribution, $\alpha \in (0,1)$ is a desired safety risk level, $\gamma_R, \gamma_C \in [0,1]$ are discount factors for rewards and costs, respectively, and H is the maximum horizon. For indexed sequences, we use the shorthand $z_{i:j} \coloneqq (z_i, \dots, z_j)$ and $z_{t,i:j} \coloneqq (z_{t,i}, \dots, z_{t,j})$, with index ranges $i > j$ denoting the empty sequence. The following includes details on our setup.

State Space $\mathcal{S}$. We consider a general state space $\mathcal{S}$, whose representation depends on the task domain. Besides ordinary task states, $\mathcal{S}$ includes a successful terminal state `TER` and a halted state `HAL`. Each episode t starts at $s_{t,1} \sim p_1$, where p_1 is supported on $\mathcal{S} \setminus \{\texttt{TER}, \texttt{HAL}\}$.

Action Space $\mathcal{A}$. The action space $\mathcal{A} \coloneqq \mathcal{A}^{\text{task}} \cup \{\texttt{INT}, \texttt{WAI}\}$ consists of task actions, intervention, and waiting. As a last-resort safety measure, `INT` replaces the proposed task action and hands control to an external supervisor or a safe recovery mechanism.

Policy. We design an *execution policy* $\pi^{\text{exe}} : \mathcal{S} \to \mathcal{A}$ that selects the executed action through action decoding, possibly using internal randomization. It may be updated from observed interaction history both between and within episodes, potentially changing the induced state-action distribution. Dependence on history and randomization is left implicit in the notation.

State Transition Kernel, Reward, and Cost. The joint kernel $p : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S} \times \mathcal{R} \times \mathcal{C})$ specifies the conditional distribution of the next state, reward, and cost given the current state and action. At ordinary task states, the admissible actions are $\mathcal{A}^{\text{task}} \cup \{\texttt{INT}\}$. Choosing `INT` moves the process to `HAL` with zero reward and unit cost, the latter assigned as a conservative convention. In `HAL`, only `WAI` is admissible; it leaves the state unchanged and yields zero reward and cost. Thus, intervention ends task execution but does not itself terminate the episode. On task completion, the process enters `TER` and the episode ends; no further action is selected.

Trajectory. Online adaptation proceeds over T episodes. Here, T counts episodes until adaptation ends and may be random, depending on the observed history. Let $t \in \{1, \dots, T\}$ index trajectories and $h \in \{1, \dots, H\}$ index within-trajectory steps. Interaction between an execution policy and an environment yields a trajectory $\tau_t \coloneqq \{(s_{t,h}, a_{t,h}, r_{t,h}, c_{t,h}, s_{t,h+1})\}_{h=1}^{H_t}$, where H_t is the *effective horizon*. It counts recorded interaction steps until task completion, horizon H, or termination of

adaptation. We also define the *task horizon* $H_t^{\text{task}} := \sum_{h=1}^{H_t} \mathbb{1}(a_{t,h} \in \mathcal{A}^{\text{task}})$, counting executed task actions, and the *active horizon* $H_t^{\text{act}} := \sum_{h=1}^{H_t} \mathbb{1}(a_{t,h} \in \mathcal{A}^{\text{task}} \cup \{\texttt{INT}\})$, counting active steps, *i.e.,* task actions or interventions. Thus, $H \geq H_t \geq H_t^{\text{act}} \geq H_t^{\text{task}}$. Given an MDP, we write $\tau \sim \pi^{\text{exe}}$ for sampling from the trajectory distribution induced by the policy.

Selective Safety Risk. As for our safety metric, we use a *selective safety risk* (SSR):

$$\mathbf{SSR}(\tau_{1:T}) := \frac{\sum_{t=1}^{T} \sum_{h=1}^{H_t} \mathbb{1}(a_{t,h} \in \mathcal{A}^{\text{task}}) \cdot c_{t,h}}{\sum_{t=1}^{T} H_t^{\text{task}}}.$$

When no task action is executed, we set $\mathbf{SSR}(\tau_{1:T}) = \alpha$ by convention. Intuitively, SSR measures the average realized safety cost over all executed task actions across adaptation episodes, and we aim to control this quantity at the desired safety-risk level α. Here, we note that SSR has the form of a realized false discovery proportion if $c_{t,h} \in \{0, 1\}$ and $\sum_{t=1}^{T} H_t^{\text{task}} > 0$.

Efficiency. A conservative execution policy may meet the safety-risk target yet allow too little task execution to be useful. We measure task-execution *efficiency* as the fraction of interaction steps executing task actions: $\mathbf{Eff}(\tau_{1:T}) := \sum_{t=1}^{T} H_t^{\text{task}} \;/\; \sum_{t=1}^{T} H_t$. This definition has two desirable properties. First, it does not penalize early success: completion without intervention gives $H_t^{\text{task}} = H_t \leq H$ and a task-execution fraction of one. Second, it accounts for intervention-induced inactivity: after intervention, the process waits in `HAL` until horizon H or adaptation ends. Intervention and recorded waiting count toward H_t but not H_t^{task}, giving $H_t^{\text{task}} < H_t \leq H$.

Goal. We seek an online adaptation algorithm Alg that updates its execution policy using the observed history: $\pi_{t,h}^{\text{exe}} = \mathsf{Alg}(\tau_{1:t-1}, \tau_{t,1:h-1})$. It should achieve a high expected average discounted task return $\mathbb{E}J_T^{\text{task}}(\tau_{1:T}) := \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\sum_{h=1}^{H_t}\gamma_R^{h-1} r_{t,h}\right]$ while satisfying, for any $\delta \in (0, 1)$,

$$\mathbb{P}\Big(\mathbf{SSR}(\tau_{1:T}) \leq \alpha + \varepsilon(\tau_{1:T}, \delta)\Big) \geq 1 - \delta, \tag{1}$$

where expectation and probability are over the interaction between $\mathcal{M}$ and Alg. We call this *adaptive safety at level* α if the prespecified, trajectory-dependent slack $\varepsilon(\tau_{1:T}, \delta) \geq 0$ is non-increasing in δ and, for each fixed δ, vanishes in probability as the realized active-step count $N^{\text{act}}(\tau_{1:T}) := \sum_{t=1}^{T} H_t^{\text{act}} \to \infty$. Section 4.3 gives sufficient conditions for our method. With bounded nonnegative task-action rewards, higher expected return raises a necessary lower bound on expected efficiency, which we do not optimize explicitly (Appendix F.1).

4 Method: Online Conformal Shielding

We propose *Online Conformal Shielding* (OCS) for safe online policy adaptation under coupled exogenous and endogenous distribution shifts. OCS compares each proposed action's safety-risk score with an adaptive threshold to decide whether to execute or intervene. Our generic reduction to online conformal abstention (OCA) (Lee et al., 2026) transfers the selective-risk guarantee of any OCA procedure satisfying Assumption 1 to execution-level safety. We develop a prior-aware OCA variant that uses pre-adaptation information to guide threshold selection.

4.1 Shielding Policy

We implement the execution policy using a *shielding policy* that decides whether to execute a proposed task action or intervene (Alshiekh et al., 2018). At each decision step, the task policy first proposes an action without executing it: $\pi_t^{\text{task}} : \mathcal{S} \to \mathcal{A}^{\text{task}}$. The proposed action is evaluated by an action-conditioned safety-risk scorer $\bar{Q}_t^{\text{cost}} : \mathcal{S} \times \mathcal{A}^{\text{task}} \to [0, 1]$, which predicts the safety risk associated with executing the proposed action at the current state. Given the current task action $a_{t,h}^{\text{task}} := \pi_t^{\text{task}}(s_{t,h})$, its safety-risk score $q_{t,h} := \bar{Q}_t^{\text{cost}}(s_{t,h}, a_{t,h}^{\text{task}})$, and a threshold $\eta_{t,h}$, the shielding policy decides whether it shields the current action or not, as follows:

$$\pi_{t,h}^{\text{sh}}(s_{t,h}; \eta_{t,h}) := \begin{cases} a_{t,h}^{\text{task}} & \text{if } s_{t,h} \neq \texttt{HAL} \text{ and } q_{t,h} \leq \eta_{t,h} \\ \texttt{INT} & \text{if } s_{t,h} \neq \texttt{HAL} \text{ and } q_{t,h} > \eta_{t,h} \\ \texttt{WAI} & \text{otherwise} \end{cases}. \tag{2}$$

Thus, the proposed task action is executed only when its predicted safety risk does not exceed the selected threshold. Otherwise, the shielding policy intervenes by choosing `INT`. Section 4.2 describes how the task policy, safety-risk scorer, and threshold are updated.

4.2 Generic Online Algorithm for Conformal Shielding

4.2.1 Online Threshold Adaptation via Online Conformal Abstention

We reduce online threshold adaptation to OCA under adversarial bandit feedback (Lee et al., 2026). To handle environment shifts and updates to the task policy and safety-risk scorer, we require an OCA algorithm with a selective-risk violation guarantee under a non-anticipating adaptive adversary, as formalized in Assumption 1.

Let $\mathcal{X}$ and $\mathcal{Y}^{\text{label}}$ denote the example and label spaces, and set $\mathcal{Y} \coloneqq \mathcal{Y}^{\text{label}} \cup \{\texttt{IDK}\}$, where IDK ("I don't know") denotes abstention. At step n, OCA updates an abstainer $\hat{S}_n : \mathcal{X} \to \mathcal{Y}$ using a time-varying risk scorer $f_n : \mathcal{X} \times \mathcal{Y}^{\text{label}} \to [0,1]$ and a threshold ξ_n. On input x_n, the abstainer returns the proposed prediction $\hat{y}_n \in \mathcal{Y}^{\text{label}}$ if $f_n(x_n, \hat{y}_n)$ does not exceed ξ_n, and returns IDK otherwise. We require the OCA algorithm to satisfy the following assumption.

Assumption 1. *An OCA algorithm* $\mathsf{Alg}_{OCA} \coloneqq (\mathsf{Infer}_{OCA}, \mathsf{Update}_{OCA})$ *adapts its parameter* ξ_n *through the following protocol with an adversarial environment: at each step* $n = 1, \ldots, N$*,*

1. *a non-anticipating adaptive adversary*[1] *chooses an example* x_n *and a label* y_n*, which determine the feedback function* $E_n : \mathcal{Y} \times \mathcal{Y} \to [0,1]$ *to measure the prediction error,*
2. *the algorithm observes* x_n *and fixes the proposed prediction* $\hat{y}_n$ *and its score* $f_n(x_n, \hat{y}_n)$*; it then selects* ξ_n *via* Infer_{OCA} *using fresh randomization and outputs* $\hat{S}_n(x_n)$*, and*
3. *the algorithm observes the proposed prediction's feedback* $e_n \coloneqq E_n(y_n, \hat{y}_n)$ *only if* $\hat{S}_n(x_n) \neq$ IDK*, and invokes* Update_{OCA} *using the available information.*

For every deterministic horizon N*, every interaction sequence generated by the above protocol with a non-anticipating adaptive adversary, and every* $\alpha, \delta \in (0,1)$*, the OCA algorithm satisfies the following selective risk violation guarantee:*

$$\mathbb{P}\left(\mathcal{V}_N \leq B_{\mathrm{OCA}}(N,\delta)\right) \geq 1-\delta, \text{ where } \mathcal{V}_N \coloneqq \sum_{n=1}^{N} \mathbb{1}(\hat{S}_n(x_n) \neq \texttt{IDK})(e_n - \alpha). \tag{3}$$

Here, $B_{\mathrm{OCA}}(N,\delta)$ *is a fixed-horizon bound that is sublinear in* N *for each fixed* δ*.*

We require an OCA guarantee against a non-anticipating adaptive adversary to handle distribution shifts during safe adaptation. As the shielding policy is updated, the distribution over future trajectories changes; this setting handles these changes without explicitly modeling that distribution.

To apply the OCA guarantee to shielding, we construct the following correspondence. We map each active OCS step (t,h) to an OCA round via $\kappa(t,h) \coloneqq \sum_{i=1}^{t-1} H_i^{\text{act}} + h$, for $h \in \{1, \ldots, H_t^{\text{act}}\}$. On active steps, identifying INT with IDK, the components correspond as follows: $\bar{Q}_t^{\text{cost}} = f_{\kappa(t,h)}$, $\pi_{t,h}^{\text{sh}} = \hat{S}_{\kappa(t,h)}$, $\eta_{t,h} = \xi_{\kappa(t,h)}$, $s_{t,h} = x_{\kappa(t,h)}$, $a_{t,h}^{\text{task}} = \hat{y}_{\kappa(t,h)}$, $\mathcal{S} = \mathcal{X}$, and $\mathcal{A}^{\text{task}} = \mathcal{Y}^{\text{label}}$. On steps where a task action is executed, $c_{t,h} = e_{\kappa(t,h)}$. OCA is updated after every active step, including interventions, but an intervention does not reveal the unexecuted task action's cost, and the assigned intervention cost is not used as feedback for that action. Based on this mapping, Section 4.3 derives an OCS selective safety risk bound from (3) and specifies conditions under which the finite-sample slack vanishes, yielding the adaptive safety guarantee (1).

4.2.2 Prior-aware Online Conformal Abstention

Safe adaptation must balance early risk control with task execution under a limited interaction budget. Overly permissive thresholds may admit high-cost actions, whereas overly conservative thresholds trigger frequent interventions. To guide early threshold selection with information available before adaptation, such as the pretrained scorer's typical safety-risk score, we replace the uniform initialization of ExAUL (Lee et al., 2026) with a fixed, full-support prior $w_1 \in \Delta(\mathcal{H})$ over a finite candidate set $\mathcal{H}$. Only the initialization changes; threshold sampling, loss estimation, and weight updates retain the original rules. Under the conditions of Theorem 2 in Appendix B.1, this variant satisfies Assumption 1 with an $\mathcal{O}(\sqrt{N})$ selective-risk violation bound for fixed $\mathcal{H}$, w_1, and δ.

4.2.3 Learning a Safety-Risk Scorer

We adapt the action-conditioned safety-risk scorer $\bar{Q}_t^{\text{cost}}$ to evaluate task-action proposals under environment and policy shifts, aiming to reduce unnecessary interventions. After episode t, we train

[1] A non-anticipating adversary chooses the current example and label using the history through step $n-1$, before OCA randomizes at step n. Given that history, these choices are independent of OCA's fresh randomness.

a cost-Q head by temporal-difference learning using only the executed task-action transitions in τ_t. Outputs are clipped to $[0, 1]$, and horizon normalization is applied to training targets, so inference does not require the eventual effective horizon. Under Assumption 1, safety does not require calibrated scores: OCA controls selective-risk violation, while the scorer's predictive quality affects execution efficiency. Appendix B.2 gives architecture and training details.

4.2.4 Constrained Reinforcement Learning for a Task Policy Adaptation

We adapt π_t^{task} through constrained reinforcement learning to improve task return and reduce interventions under environment shifts. In our implementation, PPO with an augmented Lagrangian updates the policy after episode t using only executed task-action transitions in τ_t, with $\mathcal{V}(\tau_t, \alpha) \leq 0$ as the empirical safety constraint. Under Assumption 1, the OCA guarantee underlying adaptive safety remains valid for history-dependent policy and scorer updates, without requiring this constraint to hold in every episode. Appendix B.3 gives the objective and optimization details.

4.2.5 Online Conformal Shielding Algorithm

Algorithm 1 in Appendix B combines OCA-based threshold adaptation with task-policy and safety-risk-scorer updates to form $\mathsf{Alg}_{\text{OCS}}$. It supports any OCA algorithm satisfying Assumption 1 and history-dependent policy and scorer update rules, which we collectively denote by PolicyAndScorerAdaptation. At each active step, the shielding policy decides whether to execute the proposed task action or intervene; the algorithm then executes the chosen action, observes the reward and cost, and updates OCA using the available feedback. After each episode, PolicyAndScorerAdaptation updates the task policy and safety-risk scorer from the trajectory τ_t.

A fixed active-step budget N^{act} provides a practical stopping rule by capping the total number of task-action executions and interventions during adaptation. Unlike a fixed episode count, this budget directly controls active interaction even when early task completion or intervention changes episode lengths. Waiting steps do not consume the budget or update OCA. The algorithm stops when $N^{\text{act}}(\tau_{1:T}) = N^{\text{act}}$ and returns the resulting episode count T. Since each active step corresponds to one OCA update, prescribing this budget before interaction also fixes the OCA horizon at $N = N^{\text{act}}$, allowing the fixed-horizon guarantee in Assumption 1 to apply.

4.3 Adaptive Safety Guarantee for Online Conformal Shielding

We analyze the safety of OCS under coupled environment shifts from environment changes and task-policy or safety-risk-scorer updates. Under Assumption 1, we provide a high-probability *adaptive safety guarantee* that is agnostic to the particular update rules and makes the role of task-execution efficiency explicit. The proofs of Theorem 1 and the vanishing adaptive-safety slack implication below are given in Appendix C.2.

Theorem 1 (Adaptive safety under coupled shifts)**.** *Consider* Alg_{OCS} *in Algorithm 1, instantiated with* Alg_{OCA} *as its online threshold-selection subroutine and run with a prescribed active-step budget* $N^{act} \in \mathbb{N}$ *and per-episode horizon* H*. Let* T *be the resulting number of episodes, so that* $\sum_{t=1}^{T} H_t^{act} = N^{act}$*, and let* $N^{task} := \sum_{t=1}^{T} H_t^{task}$*. If* Alg_{OCA} *satisfies Assumption 1, then, for any task-policy and safety-risk-scorer update rules in the non-anticipating adaptive-adversary setting covered by that assumption and any* $\alpha, \delta \in (0, 1)$*, with probability at least* $1 - \delta$*, whenever* $N^{task} > 0$*,*

$$\textbf{\textit{SSR}}(\tau_{1:T}) \leq \alpha + \frac{B_{\text{OCA}}(N^{act}, \delta)}{N^{task}} = \alpha + \frac{B_{\text{OCA}}(N^{act}, \delta)}{\textbf{\textit{Eff}}(\tau_{1:T}) \sum_{t=1}^{T} H_t}. \tag{4}$$

Vanishing adaptive-safety slack. When $N^{\text{task}} = 0$, $\mathbf{SSR}(\tau_{1:T}) = \alpha$ by definition. For $N^{\text{task}} > 0$, (4) gives the finite-sample slack $B_{\text{OCA}}(N^{\text{act}}, \delta)/N^{\text{task}}$. Because Assumption 1 ensures $B_{\text{OCA}}(N^{\text{act}}, \delta) = o(N^{\text{act}})$ for each fixed δ, this slack vanishes in probability as $N^{\text{act}} \to \infty$ whenever the executed-task fraction $N^{\text{task}}/N^{\text{act}}$ is bounded below by a positive constant with probability tending to one. A sufficient condition is that efficiency remains bounded away from zero with probability tending to one, since $N^{\text{task}}/N^{\text{act}} \geq \mathbf{Eff}(\tau_{1:T})$.

Implication for safety staleness. Theorem 1 shows that $\mathsf{Alg}_{\text{OCS}}$ addresses *safety staleness* under both *exogenous* environment shifts and *endogenous* policy or safety-risk-scorer updates. Our main problem formulation and experiments capture a practically relevant deployment setting: the system first encounters an exogenous shift upon entering the target environment and then adapts within that fixed target MDP, while updates to the policy or safety-risk scorer continue to induce endogenous

shifts. More generally, the theorem allows the environment to shift between any two adaptation episodes, jointly with endogenous policy or safety-risk-scorer updates. Both sources of shift may depend arbitrarily on the past interaction history and need not be stationary or explicitly modeled; this is precisely the non-anticipating adaptive-adversary setting covered by Assumption 1. Because the guarantee is not tied to a fixed calibration distribution, such shifts do not invalidate the cumulative guarantee or require safety recalibration from scratch.

Prior-aware adaptive safety guarantee. Under the conditions of Theorem 2, replacing $\mathrm{KL}(\delta_{\xi^\star}\|w_1)$ by $\ln(1/\min_{\xi\in\mathcal{H}} w_1(\xi))$ in (6) gives prior-aware `ExAUL` a deterministic bound $B_{\mathrm{OCA}}(N,\delta) = \mathcal{O}(\sqrt{N})$ for fixed $\mathcal{H}$, full-support w_1, and δ. This comparator-independent bound applies in (4) with $N = N^{\mathrm{act}}$; Appendix B.1 details its derivation and its dependence on the fixed initialization prior.

5 Experiments

We evaluate whether OCS addresses the two challenges of safe adaptation (Section 1): **(Q1) safety staleness**, *i.e.,* whether OCS keeps the selective safety risk at α under coupled exogenous and endogenous shifts, where policy-level safe RL and fixed shields fail; **(Q2) the safety–exploration imbalance**, *i.e.,* whether it does so without sacrificing execution efficiency and task success; and **(Q3)** whether α controls the resulting risk–utility trade-off. Videos of OCS during online adaptation are available at `https://iclr53876.github.io/shield-demo/`.

5.1 Setup

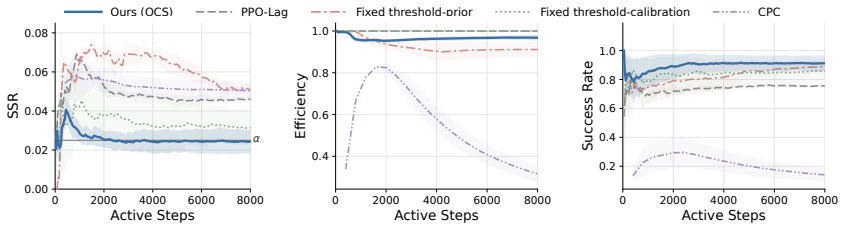
Benchmarks. We evaluate OCS on two complementary benchmarks. *ManiSkill* (Liu et al., 2025) uses a VLA manipulation policy under scene, robot-initialization, and visual shifts, with safety costs designed to capture practical manipulation failures such as unintended contact and object displacement. *Safety-Gymnasium* (Ji et al., 2023) uses a state-based locomotion policy under changes in environmental hazards. For both benchmarks, we pretrain the task policy and safety-risk scorer in-distribution using PPO-Lag and then adapt them online in each target environment. Detailed environments, safety costs, and training protocols are provided in Appendices D.2, D.3, and D.4.

Baselines. We compare OCS against **PPO-Lag**, which uses no shield; two **fixed shields**, whose thresholds are selected from the prior or offline calibration and remain unchanged during adaptation; and **CPC** (Prinster et al., 2026), a conformal policy-level method that constrains deviation from the pretrained policy. Baseline implementation details are provided in Appendices D.4 and D.5.

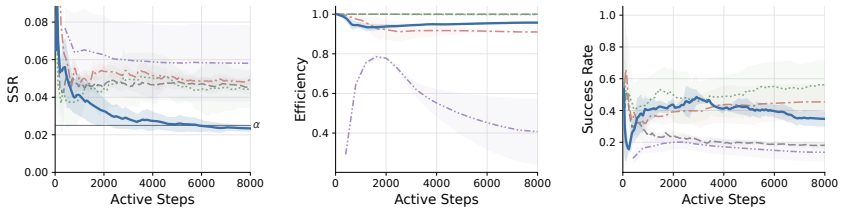
Metrics. We report three cumulative metrics: **Selective Safety Risk (SSR)**, the average safety cost over executed task actions, **Efficiency**, the fraction of interaction steps that execute task actions, and **Success Rate**, the fraction of adaptation episodes that successfully complete the task. SSR is compared against the target level α. Curves are averaged over three seeds, with shaded regions indicating standard deviation. Further evaluation and plotting details are provided in Appendices D.4 and D.5.

5.2 Q1: Safety Staleness

OCS maintains selective safety risk near the target under coupled shifts, whereas policy-level safe RL and fixed shields suffer from safety staleness. On ManiSkill (Figure 2), with target risk level $\alpha = 0.025$, the final SSR of OCS remains within $[0.018, 0.029]$ across all five environments, while every baseline finishes above the target (Table 6). The fixed-shield results directly illustrate safety staleness. Fixed threshold-calibration exceeds the target risk level across all five distribution shifted environments. Although its threshold is calibrated to the pretrained policy and scorer, both continue to change during online adaptation, causing the fixed threshold to become progressively miscalibrated under endogenous shift. This mismatch persists and can be further compounded when the deployment environment also changes. Fixed threshold-prior performs even worse: despite intervening more frequently than OCS, it incurs higher SSR and can even be less safe than PPO-Lag without shielding. These results show that simply fixing a more conservative threshold is insufficient, since a threshold calibrated to past interactions can become misaligned with the evolving policy, scorer, and environment.s Policy-level safety control does not resolve this issue either. PPO-Lag constrains risk only through policy updates after unsafe interactions have occurred, . . . while CPC also finishes above α in every ManiSkill environment, even with $12\times$ more interaction, since no safe reference policy is available to constrain it (Appendix D.5.4).

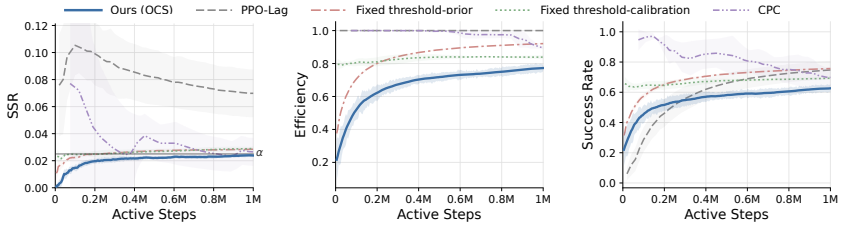


(a) `MultiCarrot`

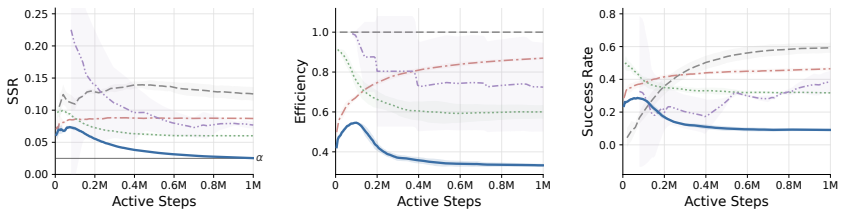


(b) `EEPose`

Figure 2: Online adaptation on ManiSkill with $\alpha = 0.025$: (a) the pretraining environment and (b) the robot-pose shift. Each row shows the cumulative SSR (horizontal line: target α), execution efficiency, and success rate against active steps, as mean $\pm$ std over three seeds; CPC's curve is compressed $12\times$ along the x-axis (Section 5.1). Results on `Position`, `VisionWhole`, and `VisionTexture` are in Appendix E.2.



(a) `SafetyPointGoal1`



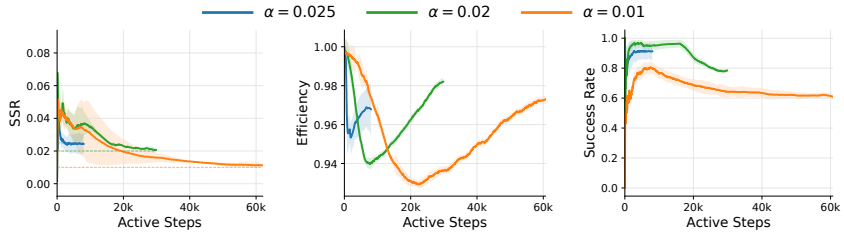
(b) `SafetyPointGoal2`

Figure 3: Online adaptation on Safety-Gymnasium with $\alpha = 0.025$: (a) the in-distribution `SafetyPointGoal1` and (b) `SafetyPointGoal2` with more hazards. Layout as in Figure 2.

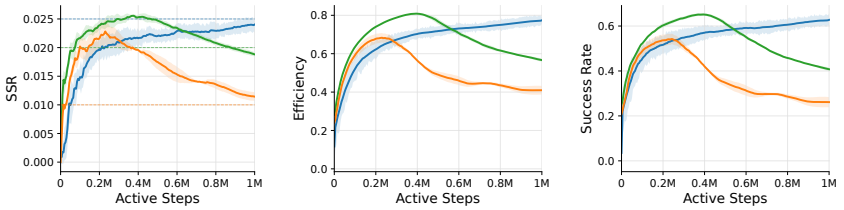
In contrast, OCS continuously recalibrates its shielding threshold from observed safety costs without requiring an explicit model of the underlying shift. Because shielding is applied before executing a proposed action, OCS can regulate safety throughout adaptation rather than relying solely on delayed policy updates. We observe the same pattern on Safety-Gymnasium (Figure 3). On `SafetyPointGoal1`, OCS reaches the target risk level while the fixed shields and PPO-Lag finish above it. The difference becomes more pronounced on the out-of-distribution `SafetyPointGoal2`, where all baselines remain at 2.4–5 $\times$ α, whereas OCS steadily reduces SSR toward the target.

5.3 Q2: Safety–Exploration Imbalance

OCS achieves this risk control while intervening only when necessary, so the task policy keeps most of its exploration. On ManiSkill, OCS keeps an execution efficiency of 0.96–0.97 in every environment, *i.e.,* it intervenes on only a few percent of the steps (Figure 2). In the in-distribution environment and under the visual shifts, OCS reaches a success rate of 0.91–0.93, on par with the best baseline and clearly above PPO-Lag (0.75–0.77). Geometric shifts are harder for



(a) ManiSkill, `MultiCarrot`



(b) Safety-Gymnasium, `SafetyPointGoal1`

Figure 4: Sensitivity of OCS to $\alpha \in \{0.025, 0.02, 0.01\}$ on (a) `MultiCarrot` and (b) `SafetyPointGoal1`. Layout as in Figure 2, with a dashed line marking each target.

all methods: on `EEPose`, OCS reaches 0.35, above PPO-Lag (0.18) but below the fixed shields (0.45–0.56), which operate at about twice the target risk, *i.e.,* at a different point of the risk–utility trade-off. When the environment itself is much riskier, meeting the target necessarily costs utility: on `SafetyPointGoal2`, where the unshielded risk is $5 \times \alpha$, OCS intervenes more often (**Eff** ≈ 0.33) rather than silently violating the target, and α governs this trade-off (Section 5.4). The threshold prior of prior-aware OCA also matters for utility rather than safety: on `SafetyPointGoal1`, replacing the prior centered at 0.9 with a uniform one still keeps SSR below α but starts from a conservative threshold that lowers efficiency and success, as predicted by Theorem 2 (Appendix E.3).

5.4 Q3: Controlling the Risk Level with α

The target risk level α reliably sets the risk–utility trade-off. We run OCS with $\alpha \in \{0.025, 0.02, 0.01\}$ on `MultiCarrot` and `SafetyPointGoal1` (Figure 4). In both benchmarks, the SSR of OCS converges toward each prescribed level, and the final values are ordered by α. A tighter target lowers task success and, on `SafetyPointGoal1` and for $\alpha = 0.01$ on `MultiCarrot`, makes the shield intervene more often; it also takes longer to settle, since the slack $B_{\mathrm{OCA}}(N^{\mathrm{act}}, \delta)/N^{\mathrm{task}}$ in Theorem 1 must become small relative to α; we therefore use active-step budgets of $8{,}000$, $32{,}000$, and $64{,}000$ for $\alpha = 0.025$, 0.02, and 0.01 on `MultiCarrot`. Practitioners can thus set the balance between safety and exploration explicitly, while OCS maintains the chosen level.

6 Conclusion

We proposed online conformal shielding (OCS) for safe policy adaptation, which frames adaptation as a pre-execution selective decision problem. The task policy proposes an action, a safety-risk scorer evaluates it, and an online-learned conformal threshold determines whether the action is executed or replaced by an intervention. By updating this threshold from partial feedback, OCS controls the selective safety risk of executed actions under coupled distribution shifts while maintaining execution efficiency for constrained RL updates. Our experiments on VLA manipulation and Safety-Gymnasium show that OCS keeps the selective safety risk near the target level while preserving useful task performance.

Limitations. We consider intervention as a safe backup strategy, like conventional shielding methods; designing a valid backup strategy is an important future direction. Also, the threshold is calibrated on the immediate per-step cost, whereas the safety-risk scorer estimates long-term cost; the guarantee holds regardless, but aligning the two could reduce unnecessary interventions.

7 AI USE STATEMENT

We used generative AI tools to help implement experiment and plotting scripts, give feedback on ablation design, and interpret results, as well as to draft parts of the paper, to edit figures, code, and text for readability and consistency, and to flag places needing references. They were not used to develop the theory, formulate mathematical claims, write proofs, or generate or process datasets. The authors verified all AI-assisted work, including every reported number and all code, and take full responsibility for the content of this paper.

REFERENCES

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

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.

Arko Banerjee, Kia Rahmani, Joydeep Biswas, and Isil Dillig. Dynamic model predictive shielding for provably safe reinforcement learning. *Advances in Neural Information Processing Systems*, 37: 100131–100159, 2024.

Osbert Bastani. Safe reinforcement learning with nonlinear dynamics via model predictive shielding. In *2021 American control conference (ACC)*, pp. 3488–3494. IEEE, 2021.

Archana Bura, Aria HasanzadeZonuzy, Dileep Kalathil, Srinivas Shakkottai, and Jean-Francois Chamberland. Dope: Doubly optimistic and pessimistic exploration for safe reinforcement learning, 2022. URL https://arxiv.org/abs/2112.00885.

Yuhui Chen, Shuai Tian, Shugao Liu, Yingting Zhou, Haoran Li, and Dongbin Zhao. Conrft: A reinforced fine-tuning method for vla models via consistency policy. *arXiv preprint arXiv:2502.05450*, 2025.

Dongsheng Ding, Xiaohan Wei, Zhuoran Yang, Zhaoran Wang, and Mihailo R. Jovanović. Provably efficient safe exploration via primal-dual policy optimization, 2020. URL https://arxiv.org/abs/2003.00534.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. *Advances in neural information processing systems*, 30, 2017.

Qiao Gu, Yuanliang Ju, Shengxiang Sun, Igor Gilitschenski, Haruki Nishimura, Masha Itkina, and Florian Shkurti. Safe: Multitask failure detection for vision-language-action models. *arXiv preprint arXiv:2506.09937*, 2025.

Songqiao Hu, Zeyi Liu, Shuang Liu, Jun Cen, Zihan Meng, and Xiao He. Vlsa: Vision-language-action models with plug-and-play safety constraint layer. *arXiv preprint arXiv:2512.11891*, 2025.

Jiaming Ji, Borong Zhang, Jiayi Zhou, Xuehai Pan, Weidong Huang, Ruiyang Sun, Yiran Geng, Yifan Zhong, Josef Dai, and Yaodong Yang. Safety gymnasium: A unified safe reinforcement learning benchmark. *Advances in Neural Information Processing Systems*, 36:18964–18993, 2023.

Minjae Kwon, Tyler Ingebrand, Ufuk Topcu, and Lu Feng. Runtime safety through adaptive shielding: From hidden parameter inference to provable guarantees. *arXiv preprint arXiv:2506.11033*, 2025.

Minjae Lee, Kyungmin Kim, Taesoo Kim, and Sangdon Park. Selective generation for controllable language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=glfYOAzh2f.

Minjae Lee, Yoonjae Jung, and Sangdon Park. Online conformal abstention under adversarial bandit feedback. In *The Fortieth Annual Conference on Neural Information Processing Systems*, 2026. URL https://arxiv.org/abs/2506.14067.

Shuo Li and Osbert Bastani. Robust model predictive shielding for safe reinforcement learning with stochastic dynamics. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7166–7172. IEEE, 2020.

Jijia Liu, Feng Gao, Bingwen Wei, Xinlei Chen, Qingmin Liao, Yi Wu, Chao Yu, and Yu Wang. What can rl bring to vla generalization? an empirical study. *arXiv preprint arXiv:2505.19789*, 2025.

Guanxing Lu, Wenkai Guo, Chubin Zhang, Yuheng Zhou, Haonan Jiang, Zifeng Gao, Yansong Tang, and Ziwei Wang. Vla-rl: Towards masterful and general robotic manipulation with scalable reinforcement learning. *arXiv preprint arXiv:2505.18719*, 2025.

Thomas M. McDonald, Kaylee Champion, Francesco Stradi, and Niklas Kochdumper. Safe exploration via policy priors. *arXiv preprint arXiv:2601.19612*, 2026.

Sangdon Park, Osbert Bastani, Nikolai Matni, and Insup Lee. Pac confidence sets for deep neural networks via calibrated prediction. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJxVI04YvB.

Drew Prinster, Clara Fannjiang, Ji Won Park, Kyunghyun Cho, Anqi Liu, Suchi Saria, and Samuel Stanton. Conformal policy control. *arXiv preprint arXiv:2603.02196*, 2026.

Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning.(2019). *URL https://cdn. openai. com/safexp-short. pdf*, 3:14–28, 2019.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer Science & Business Media, 2005.

Nolan Wagener, Byron Boots, and Ching-An Cheng. Safe reinforcement learning using advantage-based intervention, 2021. URL https://arxiv.org/abs/2106.09110.

Nuoya Xiong, Yihan Du, and Longbo Huang. Provably safe reinforcement learning with step-wise violation constraints, 2023. URL https://arxiv.org/abs/2302.06064.

Yifan Yang, Zhixiang Duan, Tianshi Xie, Fuyu Cao, Pinxi Shen, Peili Song, Piaopiao Jin, Guokang Sun, Shaoqing Xu, Yangwei You, et al. Fpc-vla: A vision-language-action framework with a supervisor for failure prediction and correction. *arXiv preprint arXiv:2509.04018*, 2025.

Borong Zhang, Yuhao Zhang, Jiaming Ji, Yingshan Lei, Josef Dai, Yuanpei Chen, and Yaodong Yang. Safevla: Towards safety alignment of vision-language-action model via constrained learning. *arXiv preprint arXiv:2503.03480*, 2025.

Hao Zhou, Yanze Zhang, and Wenhao Luo. Computationally and sample efficient safe reinforcement learning using adaptive conformal prediction. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 01–07. IEEE, 2025.

A Additional Related Work

Table 1: Comparison of assumptions for safety guarantees on representative methods.

Method	Assumptions for Safety				Distribution Shifts	
	State/Action Setting	Dynamics-Agnostic	Cost-Function-Agnostic	Initial-Safe-Policy-Free	Endogenous Shift[c]	Exogenous Shift[d]
Policy-level safe RL and intervention-based methods						
OPDOP (Ding et al., 2020)	Infinite	✗	✗	✓	✓	✗
SAILR (Wagener et al., 2021)	Infinite	▲[a]	✓	✓	✓	✗
DOPE (Bura et al., 2022)	Finite	▲[b]	✓	✗	✓	✗
SUCBVI (Xiong et al., 2023)	Finite	▲[b]	✓	✓	✓	✗
CPC (Prinster et al., 2026)	Infinite	✓	✓	✗	✓	✗
Runtime safety control and shielding						
Safe RL via Shielding (Alshiekh et al., 2018)	Finite	✗	✓	✓	✓	✗
MPS (Bastani, 2021)	Infinite	✗	✓	✓	✓	✗
DMPS (Banerjee et al., 2024)	Infinite	✗	✓	✓	✓	✗
OCS (ours)	Infinite	✓	✓	✓	✓	✓

▲[a]: No known transition model is required; the safety guarantee depends on critic admissibility under the true dynamics.
▲[b]: Arbitrary unknown transition kernels within finite (tabular) state–action spaces.
[c]: A check for *Endogenous Shift* means that safety guarantees remain valid under distribution shifts induced by online task-policy updates in a fixed environment.
[d]: A check for *Exogenous Shift* means that guarantees remain valid after deployment to a changed environment without requiring a new accurate dynamics model or re-establishing a dynamics-dependent safety certificate.

Conformal Safety Control. Conformal prediction and risk control provide coverage or risk guarantees under distributional assumptions (Vovk et al., 2005), while selective prediction abstains on untrusted outputs (Geifman & El-Yaniv, 2017; Lee et al., 2024). Online conformal abstention (OCA) (Lee et al., 2026) updates thresholds from partial feedback without offline calibration and allows time-varying risk scores. PAC trajectory confidence sets (Park et al., 2020) provide offline coverage under matched calibration and deployment distributions; coverage alone does not certify safe execution. Conformal policy control (CPC) (Prinster et al., 2026) uses a safe reference policy and computable policy likelihood ratios to support safe policy improvement under endogenous shifts without a dynamics model, but does not automatically cover exogenous shifts. Adaptive conformal prediction has also been combined with control barrier functions for model-based safe RL (Zhou et al., 2025). Under Assumption 1, OCS integrates OCA into policy adaptation to bound the realized average safety cost of executed task actions with high probability up to finite-sample slack (Theorem 1), without a safe reference task policy, policy likelihood ratios, or offline calibration.

B Details on Online Conformal Shielding

This appendix provides the algorithmic and implementation details of OCS, covering prior-aware threshold adaptation, safety-risk scorer learning, and constrained task-policy adaptation. Algorithm 1 summarizes the full OCS loop. OCA is updated after each active step using the available feedback, while the task policy and safety-risk scorer are updated at episode end. Waiting steps neither update OCA nor consume the prescribed active-step budget N^{act}.

Algorithm 1 Online Conformal Shielding

Require: initial policy π_1^{task} and scorer $\bar{Q}_1^{\text{cost}}$; risk level α; horizon H; active-step budget N^{act}

procedure OCS(π_1^{task}, $\bar{Q}_1^{\text{cost}}$, α, H, N^{act})
  $n \leftarrow 1$
  for $t = 1, 2, \ldots$ **while** $n \leq N^{\text{act}}$ **do**
    $\tau_t \leftarrow ()$
    for $h = 1$ to H **while** $n \leq N^{\text{act}}$ **do**
      Observe $s_{t,h}$
      // 1. Choose an action using the shielding policy $\pi_{t,h}^{\text{sh}}$
      if $s_{t,h} \neq$ `HAL` **then**
        $a_{t,h}^{\text{task}} \leftarrow \pi_t^{\text{task}}(s_{t,h})$; $q_{t,h} \leftarrow \bar{Q}_t^{\text{cost}}(s_{t,h}, a_{t,h}^{\text{task}})$
        $\eta_{t,h} \leftarrow \mathsf{Infer}_{\text{OCA}}(q_{t,h}; \alpha, N^{\text{act}})$
        $a_{t,h} \leftarrow a_{t,h}^{\text{task}}$ **if** $q_{t,h} \leq \eta_{t,h}$ **else** `INT`
      else
        $a_{t,h} \leftarrow$ `WAI`
      // 2. Execute an action
      Execute $a_{t,h}$
      Observe the next state $s_{t,h+1}$, reward $r_{t,h}$, and cost $c_{t,h}$
      Append $(s_{t,h}, a_{t,h}, r_{t,h}, c_{t,h}, s_{t,h+1})$ to τ_t
      // 3. Adapt a shielding threshold, remain in `HAL`, or terminate the t-th adaptation
      if $s_{t,h} \neq$ `HAL` **then**
        Update OCA status via $\mathsf{Update}_{\text{OCA}}(q_{t,h}, \eta_{t,h}, c_{t,h}; \alpha, N^{\text{act}})$
        $n \leftarrow n + 1$
      if $s_{t,h+1} =$ `TER` **then break**
    // 4. Adapt a safety-risk scorer and task policy
    $\pi_{t+1}^{\text{task}}, \bar{Q}_{t+1}^{\text{cost}} \leftarrow \mathsf{PolicyAndScorerAdaptation}(\pi_t^{\text{task}}, \bar{Q}_t^{\text{cost}}, \tau_t)$
  return $T \leftarrow t - 1$

B.1 Prior-aware Online Conformal Abstention

Online conformal abstention (OCA) (Lee et al., 2026) adapts an acceptance threshold using partial feedback. At each round, the proposed output and its safety-risk score q are fixed before the learner selects a threshold ξ; the output is accepted if $q \leq \xi$ and replaced by `IDK` otherwise. The proposed output's error is observed only on acceptance, but the learner updates using the available information after either outcome. Assumption 1 specifies the selective-risk violation guarantee required by OCS under a non-anticipating adaptive adversary.

Prior-aware initialization. Our extension incorporates information available before adaptation into the initial distribution over thresholds. We instantiate OCA with `ExAUL` and replace its uniform initial distribution over thresholds with a fixed, full-support prior $w_1 \in \Delta(\mathcal{H})$, which can reflect the pretrained scorer's typical safety-risk score. Only the initialization changes: threshold sampling, loss estimation, and subsequent weight updates retain the original rules. Theorem 2 gives a prior-dependent selective-risk violation bound under its stated conditions.

Inference and update. Algorithm 2 implements the interface $(\mathsf{Infer}_{\text{OCA}}, \mathsf{Update}_{\text{OCA}})$ using the feedback-unlocked update of Lee et al. (2026, Algorithms 1–2). Our acceptance rule $q \leq \xi$ uses safety-risk scores, reversing the threshold direction used for conformity scores in the original OCA formulation. The parameters and weights are initialized once; the two procedures share the current weights w and the most recently used sampling distribution p.

Algorithm 2 Prior-aware ExAUL for the OCA Interface

// For Theorem 2: $\lambda = \sqrt{N}$, $\nu = 2\gamma = \sqrt{\ln|\mathcal{H}|/N}$ (fixed horizon N)

Require: finite threshold set $\mathcal{H}$; prior $w_1 \in \Delta(\mathcal{H})$ with $w_1(\xi) > 0$ for all $\xi \in \mathcal{H}$; $\alpha \in (0,1)$; loss weight $\lambda > 0$; learning rate $\nu > 0$; exploration $\gamma > 0$

// once, before interaction; both procedures share w and the latest sampling distribution p
Initialize $w(\xi) \leftarrow w_1(\xi)$ for all $\xi \in \mathcal{H}$
Initialize $p(\xi) \leftarrow w_1(\xi)$ for all $\xi \in \mathcal{H}$

procedure $\mathsf{Infer}_{\mathrm{OCA}}(q)$
    $p(\xi) \leftarrow w(\xi)/\sum_{\zeta\in\mathcal{H}} w(\zeta)$ for all $\xi \in \mathcal{H}$
    Draw $\xi \sim p$ using fresh randomization
    return ξ

procedure $\mathsf{Update}_{\mathrm{OCA}}(q, \xi, e)$
    if $q \le \xi$ **then**
        // the proposed output was accepted
        $\mathcal{U} \leftarrow \{\zeta \in \mathcal{H} : q \le \zeta\}$
        $\ell^{\mathrm{obs}} \leftarrow \lambda e/(1+\lambda)$
    else
        // abstention: e is not used
        $\mathcal{U} \leftarrow \{\zeta \in \mathcal{H} : \zeta < q\}$
        $\ell^{\mathrm{obs}} \leftarrow (1+\lambda\alpha)/(1+\lambda)$
    for all $\zeta \in \mathcal{H}$ **do**
        $\tilde{\ell}(\zeta) \leftarrow \ell^{\mathrm{obs}}\mathbb{1}\{\zeta \in \mathcal{U}\}/(\gamma + \sum_{\zeta'\in\mathcal{U}} p(\zeta'))$
        $w(\zeta) \leftarrow w(\zeta)\exp(-\nu\tilde{\ell}(\zeta))$

Feedback unlocking. The unlocked set $\mathcal{U}$ contains exactly the candidates that make the same decision as the sampled threshold. Their losses coincide, so the feedback-unlocked estimator of Lee et al. (2026) reduces to the expression in Algorithm 2, using the pre-update sampling probabilities p. On acceptance, $e \in [0,1]$ is the observed feedback for the proposed output. On abstention, its counterfactual feedback is unavailable and the feedback argument e may be $\perp$: the loss is $(1+\lambda\alpha)/(1+\lambda)$ regardless of e.

Use in OCS. At OCA round $n = \kappa(t,h)$, corresponding to active step (t,h), Algorithm 1 calls $\mathsf{Infer}_{\mathrm{OCA}}(q_{t,h})$ to obtain $\eta_{t,h} = \xi_n$, executes or intervenes, and then calls $\mathsf{Update}_{\mathrm{OCA}}(q_{t,h}, \eta_{t,h}, c_{t,h})$. The cost $c_{t,h}$ is used only if the task action was executed; an intervention's cost is not treated as feedback for the unexecuted task action. The OCA state is retained across episodes, and neither procedure is called on waiting steps.

Prior-dependent weights. After n updates, $w_{n+1}(\xi) = w_1(\xi)\exp(-\nu\sum_{m=1}^{n}\tilde{\ell}_m(\xi))$, so w_1 remains a multiplicative factor throughout learning. The choice $w_1(\xi) = 1/|\mathcal{H}|$ recovers the original ExAUL update; a nonuniform prior changes the initial preference among thresholds.

OCA loss and regret. We use regret to quantify how the prior affects the selective-risk violation bound. Let N be a deterministic horizon, $K = |\mathcal{H}|$, and ξ_n the threshold selected at OCA round n. The true OCA loss $\ell_n(\xi)$ in (19) equals $\lambda e_n/(1+\lambda)$ when threshold ξ accepts the proposed output and $(1+\lambda\alpha)/(1+\lambda)$ when it abstains. Here $e_n = E_n(y_n, \hat{y}_n)$ is the proposed prediction's potential feedback from Assumption 1; the loss is defined even when this feedback is not observed. External regret measures the excess cumulative loss over the best fixed threshold on the same realized loss sequence:

$$\mathbf{Reg}_N \coloneqq \sum_{n=1}^{N}\ell_n(\xi_n) - \min_{\xi\in\mathcal{H}}\sum_{n=1}^{N}\ell_n(\xi). \tag{5}$$

The threshold $\xi^* \in \arg\min_{\xi\in\mathcal{H}}\sum_{n=1}^{N}\ell_n(\xi)$ is called the comparator; it is chosen in hindsight only for analysis, whereas w_1 is fixed before interaction.

Prior-dependent selective-risk guarantee. The following theorem gives the selective-risk violation bound for prior-aware ExAUL; its proof is provided in Appendix C.1.

Theorem 2 (Prior-dependent Selective-risk Violation Bound)**.** *Let $\mathcal{H}$ be a finite candidate set with $K \coloneqq |\mathcal{H}| \geq 2$ that includes an always-abstaining candidate, and fix $w_1 \in \Delta(\mathcal{H})$ with $w_1(\xi) > 0$ for every $\xi \in \mathcal{H}$. For a deterministic horizon $N \geq 1$ and $\alpha, \delta \in (0,1)$, run* ExAUL *under the non-anticipating interaction protocol of Assumption 1, with initialization w_1, learning rate $\nu = 2\gamma = \sqrt{\ln K / N}$, and loss parameter $\lambda = \sqrt{N}$. Then, with probability at least $1-\delta$,*

$$\mathcal{V}_N \leq \sqrt{N} + \left(1 + \frac{1}{\sqrt{N}}\right)\left(3\sqrt{N \ln K} + \sqrt{\frac{N}{\ln K}}\,\mathrm{KL}(\delta_{\xi^\star} \| w_1) + \left(1 + \sqrt{\frac{N}{\ln K}}\right)\ln\frac{2}{\delta}\right) = \mathcal{O}(\sqrt{N}), \quad (6)$$

where $\mathcal{V}_N$ is defined in (3)*, $\xi^\star \in \arg\min_{\xi \in \mathcal{H}} \sum_{n=1}^{N} \ell_n(\xi)$ for the OCA losses in* (19)*,* $\mathrm{KL}(\delta_{\xi^\star} \| w_1) = \ln(1/w_1(\xi^\star))$*, and the asymptotic rate holds for fixed $\mathcal{H}$, w_1, and δ.*

Effect of the prior on regret. The initialization changes the regret bound through the comparator penalty: the uniform-prior term $\ln K / \nu$ is replaced by $\mathrm{KL}(\delta_{\xi^\star} \| w_1)/\nu = \ln(1/w_1(\xi^\star))/\nu$, where ν is the learning rate. Under the conditions of Theorem 2, the analysis in Appendix C.1 gives, with probability at least $1-\delta$,

$$\mathbf{Reg}_N \leq 3\sqrt{N \ln K} + \sqrt{\frac{N}{\ln K}}\,\mathrm{KL}(\delta_{\xi^\star} \| w_1) + \left(1 + \sqrt{\frac{N}{\ln K}}\right)\ln\frac{2}{\delta}. \quad (7)$$

From regret to selective risk. The always-abstaining candidate gives the conversion (20). With $\lambda = \sqrt{N}$, this reads $\mathcal{V}_N \leq \sqrt{N} + \left(1 + \frac{1}{\sqrt{N}}\right)\mathbf{Reg}_N$. Substituting (7) yields (6) on the same probability event.

Interpreting the bound. The uniform prior gives $\mathrm{KL}(\delta_{\xi^\star} \| w_1) = \ln K$. Substituting this identity into (7) recovers the original ExAUL regret bound, with leading term $4\sqrt{N \ln K}$. The conversion in (20) then recovers the corresponding selective-risk violation bound. For a given loss sequence, the prior-dependent term is smaller precisely when $w_1(\xi^\star) > 1/K$. This compares terms in the bound, not the realized selective risks of two runs; an informative prior does not guarantee a lower realized risk. To obtain the deterministic bound required by Assumption 1, use $\mathrm{KL}(\delta_{\xi^\star} \| w_1) \leq \ln(1/\min_{\xi \in \mathcal{H}} w_1(\xi))$ in (6). Full support makes this upper bound finite. For fixed $\mathcal{H}$, w_1, and δ, the resulting $B_{\mathrm{OCA}}(N, \delta)$ is $\mathcal{O}(\sqrt{N}) = o(N)$. Thus, under the theorem's conditions, prior-aware ExAUL satisfies Assumption 1.

Conditions for the fixed-horizon bound. Applying (6) requires the conditions of Theorem 2, including a deterministic horizon of N OCA updates, $\lambda = \sqrt{N}$, $\nu = 2\gamma = \sqrt{\ln|\mathcal{H}|/N}$, and a fixed, full-support prior over a candidate set containing an always-abstaining threshold. The grid in Appendix D.4 satisfies this candidate requirement: since $q \in [0,1]$, the threshold $\xi = -0.001$ always abstains under the rule $q \leq \xi$. The exponential prior assigns positive mass to this finite threshold without a separate adjustment.

B.2 Learning a Safety-Risk Scorer

We instantiate the safety-risk scorer $\bar{Q}_t^{\text{cost}}$ of Section 4.2.3 as an action-conditioned cost-Q function that shares the backbone of the task policy.

Architecture. Given a state $s_{t,h}$, the task policy proposes $a_{t,h}^{\text{task}} = \pi_t^{\text{task}}(s_{t,h})$ as a sequence of discrete action tokens. We concatenate the final-layer hidden states of these action tokens and feed them to a lightweight MLP head, which outputs a scalar $\hat{Q}_C(s_{t,h}, a_{t,h}^{\text{task}})$. Because the head reads the hidden states of the proposed action, a single forward pass of the task policy yields both the proposal and its score.

Training objective. The cost-Q head is trained on the executed transitions $\mathcal{D}_t$ (Eq. (11)), i.e., on the same shielded data used for task-policy adaptation, since the safety cost of an intervened action is never observed. Training takes place after the trajectory τ_t has ended (step 4 of Algorithm 1), so the effective horizon H_t is known at this point. We therefore normalize the one-step temporal-difference target by the remaining effective horizon,

$$y_{t,h} \coloneqq \frac{c_{t,h} + \gamma_C\big(1 - \mathbb{I}(s_{t,h+1} = \texttt{TER})\big)\,\widehat{V}_C(s_{t,h+1})}{H_t - h + 1}, \quad (8)$$

where $\widehat{V}_C$ is the cost-value estimator trained jointly for task-policy adaptation (Appendix B.3). With this target, $\hat{Q}_C(s_{t,h}, a^{\text{task}}_{t,h})$ estimates the average safety cost per remaining step of executing $a^{\text{task}}_{t,h}$ at $s_{t,h}$, which is on the same per-step scale as the risk level α. We minimize the Huber loss

$$\mathcal{L}^{\text{cost}}(\bar{Q}^{\text{cost}}_t, \tau_t) \coloneqq \frac{1}{|\mathcal{D}_t|} \sum_{h:\, a_{t,h} \in \mathcal{A}^{\text{task}}} \text{Huber}\big(\hat{Q}_C(s_{t,h}, a^{\text{task}}_{t,h}) - y_{t,h}\big). \tag{9}$$

The same objective is used during pretraining and online adaptation.

Risk score. At decision time, the score is the clipped output of the head,

$$q_{t,h} = \bar{Q}^{\text{cost}}_t(s_{t,h}, a^{\text{task}}_{t,h}) \coloneqq \big[\hat{Q}_C(s_{t,h}, a^{\text{task}}_{t,h})\big]^1_0, \tag{10}$$

where $[z]^1_0 \coloneqq \min\{\max\{z, 0\}, 1\}$. The effective horizon H_t is unknown when the decision is made, since it depends on whether and when the shield intervenes later in the trajectory. Because the horizon normalization is absorbed into the training target, computing $q_{t,h}$ requires neither H_t nor the maximum horizon H, and the score depends only on the current state and the proposed action, as required by the shielding policy (2). The clipping keeps the score in the threshold domain $\mathcal{H} \subseteq [0, 1]$, so a single threshold grid serves all decision steps. As discussed in Section 4.2.3, $q_{t,h}$ need not be a calibrated estimate of the risk, because the threshold is calibrated online by OCA.

B.3 Constrained Reinforcement Learning for a Task Policy Adaptation

We provide the optimization details of the task-policy update in Section 4.2.4. The update is sequential: after the t-th adaptation episode, the update rule PolicyAndScorerAdaptation maps $(\pi^{\text{task}}_t, \tau_t)$ to π^{task}_{t+1} using only the data observed so far, so no expectation over future episodes is involved.[2]

Update data. At round t, we use only the executed transitions of τ_t,

$$\mathcal{D}_t \coloneqq \left\{ \left(s_{t,h}, a^{\text{task}}_{t,h}, r_{t,h}, c_{t,h}, s_{t,h+1}\right) : a_{t,h} \in \mathcal{A}^{\text{task}} \right\}, \tag{11}$$

so that $|\mathcal{D}_t| = H^{\text{task}}_t$ and the executed action in each transition coincides with the task-policy proposal. Both the intervened step ($a_{t,h} = \texttt{INT}$) and the subsequent `WAI` steps in `HAL` are excluded, and no action is resampled.

Per-round constrained problem. At round t, the update improves the task return J^{task}_t subject to the constraint $J^{\text{cost}}_t(\tau_{1:t}) = \mathcal{V}(\tau_t, \alpha) \le 0$ of Section 4.2.4. In terms of the selective safety risk of the round,

$$\mathbf{SSR}(\tau_t) = \frac{1}{|\mathcal{D}_t|} \sum_{h:\, a_{t,h} \in \mathcal{A}^{\text{task}}} c_{t,h}, \tag{12}$$

the constraint reads $\mathbf{SSR}(\tau_t) \le \alpha$, with violation

$$v_t \coloneqq \big[\mathbf{SSR}(\tau_t) - \alpha\big]_+, \qquad [z]_+ \coloneqq \max(z, 0). \tag{13}$$

Normalizing by $|\mathcal{D}_t| = H^{\text{task}}_t$ rather than by H places the constraint on the same per-action scale as SSR. Indeed,

$$\mathcal{V}(\tau_t, \alpha) = \sum_{h:\, a_{t,h} \in \mathcal{A}^{\text{task}}} (c_{t,h} - \alpha) = |\mathcal{D}_t|\big(\mathbf{SSR}(\tau_t) - \alpha\big) \tag{14}$$

is exactly the contribution of round t to the cumulative selective-risk violation controlled by OCS (Theorem 1). Reducing v_t therefore directly reduces the selective-risk violation of the executed task-policy actions.

Role of γ_C. The constraint is undiscounted, which preserves its exact correspondence with SSR; the discount factor γ_C enters only through temporal credit assignment. With the cost-value estimator $\widehat{V}_C$, we compute the cost temporal-difference residual and the cost advantage

$$\varepsilon^C_{t,h} = c_{t,h} + \gamma_C \widehat{V}_C(s_{t,h+1}) - \widehat{V}_C(s_{t,h}), \qquad A^C_{t,h} = \sum_{\ell \ge 0} (\gamma_C \lambda_{\text{GAE}})^\ell \, \varepsilon^C_{t,h+\ell}, \tag{15}$$

[2] On Safety-Gymnasium, we pool the executed transitions of 100 consecutive adaptation episodes into one update; the update itself is unchanged.

Here, the bootstrap term is zero at TER; the reward advantage $A^R_{t,h}$ is constructed analogously with γ_R. Through $A^C_{t,h}$, a safety cost incurred later in the trajectory is credited to the earlier task-policy actions that led to it. Lowering the cost advantage thus steers the task policy away from actions that cause future violations, which is how the policy update reduces the per-round violation v_t and, with it, the selective-risk violation.

Augmented-Lagrangian update. We solve the per-round problem with PPO and an augmented Lagrangian. With the Lagrange multiplier $\mu_t \geq 0$ and the penalty coefficient $\rho > 0$, we define the effective multiplier $\mu_t^{\text{eff}} := \mu_t + \rho\, v_t$ and combine the advantages of each transition in $\mathcal{D}_t$ as

$$\widetilde{A}_{t,h} = \frac{A^R_{t,h} - \mu_t^{\text{eff}} A^C_{t,h}}{1 + \mu_t^{\text{eff}}}. \tag{16}$$

Using $\widetilde{A}_{t,h}$ in the PPO clipped surrogate gives the loss $\mathcal{L}^{\text{task}}(\pi_t^{\text{task}}, \tau_t)$, which updates π_t^{task} to π_{t+1}^{task}. This update corresponds to ascending the per-round augmented Lagrangian

$$\mathcal{L}_{\text{aug},t}(\pi, \mu_t) = J_t^{\text{task}}(\pi) - \mu_t\big(\mathbf{SSR}_t(\pi) - \alpha\big) - \frac{\rho}{2}\big[\mathbf{SSR}_t(\pi) - \alpha\big]_+^2, \tag{17}$$

where $\mathbf{SSR}_t(\pi)$ denotes the round-t selective risk as a function of the task policy; its gradient with respect to the cost term carries the weight μ_t^{eff}. After the policy update, the multiplier is updated once from the observed constraint value,

$$\mu_{t+1} = \Pi_{[0,\mu_{\max}]}\Big(\mu_t + \rho\big(\mathbf{SSR}(\tau_t) - \alpha\big)\Big), \tag{18}$$

where $\Pi_{[0,\mu_{\max}]}$ is the projection onto the feasible range of the multiplier. Thus, μ_t grows while the executed task-policy actions exceed the target and shrinks otherwise, and both the task policy and its multiplier are updated sequentially from the data collected under the current shielding policy.

C Proofs

C.1 Proof of Theorem 2

Proof outline. We adapt the ExAUL analysis in the proof of Theorem 5 of Lee et al. (2026) to a general full-support prior. We explicitly state the inequalities used from that analysis and derive the prior-dependent potential bound below. The concentration inequalities do not require uniform initialization and remain applicable under the non-anticipating protocol in Assumption 1. Finally, the original regret-to-SR conversion yields the selective-risk violation bound.

Setup. For the deterministic horizon N, let $a_n(\xi)$ indicate abstention at threshold ξ and let $e_n = E_n(y_n, \hat{y}_n)$ be the potential feedback in Assumption 1. The normalized OCA losses are

$$\ell_n(\xi) := \frac{a_n(\xi) + \lambda\big((1 - a_n(\xi))(e_n - \alpha) + \alpha\big)}{1 + \lambda} \in [0, 1]. \tag{19}$$

Use $\mathbf{Reg}_N$ from (5) and choose the true-loss minimizer $\xi^\star \in \arg\min_{\xi\in\mathcal{H}} \sum_{n=1}^N \ell_n(\xi)$. Let $\tilde{\ell}_n(\xi)$ be the estimator in Algorithm 2 and $\tilde{L}_n(\xi) = \sum_{m=1}^n \tilde{\ell}_m(\xi)$, with $\tilde{L}_0 \equiv 0$. Using a nonuniform prior changes only the initial weights; the loss estimator and feedback-unlocking mechanism remain unchanged. Let $\mathcal{G}_n$ be the sigma-algebra generated by the interaction history through round $n-1$ and the current example, proposed prediction, score, and potential feedback e_n, before ξ_n is drawn. Under Assumption 1, the sampling distribution, loss vector, and feedback partition are $\mathcal{G}_n$-measurable, and fresh randomization gives

$$\Pr(\xi_n = \xi \mid \mathcal{G}_n) = p_n(\xi), \qquad \xi \in \mathcal{H}.$$

The information in $\mathcal{G}_n$ is used only for analysis; the learner need not observe the potential feedback on abstention. Conditioning on $\mathcal{G}_n$ therefore permits the original concentration arguments.

Prior-dependent step. The general-prior update has the form

$$p_n(\xi) = \frac{w_1(\xi)\exp(-\nu\tilde{L}_{n-1}(\xi))}{Z_{n-1}}, \qquad Z_n = \sum_{\xi\in\mathcal{H}} w_1(\xi)\exp(-\nu\tilde{L}_n(\xi)), \qquad Z_0 = 1.$$

Let $\mathcal{U}_n$ be the sampled feedback group in Algorithm 2 at round n. The estimator vanishes outside $\mathcal{U}_n$, and losses coincide within this group, so

$$\mathbb{E}_{\xi\sim p_n}[\tilde{\ell}_n(\xi)^2] = \frac{\ell_n(\xi_n)\sum_{\xi\in\mathcal{U}_n} p_n(\xi)}{\gamma + \sum_{\xi\in\mathcal{U}_n} p_n(\xi)}\tilde{\ell}_n(\xi_n) \le \tilde{\ell}_n(\xi_n).$$

Here the expectation is the p_n-weighted average of the realized estimator vector. The inequality uses $\ell_n(\xi_n) \le 1$, not $\tilde{\ell}_n(\xi_n) \le 1$. Applying $e^{-x} \le 1 - x + x^2/2$ for $x \ge 0$ and $\ln z \le z - 1$ for $z > 0$ to Z_n/Z_{n-1}, followed by this second-moment bound, gives the original potential argument for constant ν:

$$\begin{aligned}
\sum_{n=1}^{N} \mathbb{E}_{\xi\sim p_n}[\tilde{\ell}_n(\xi)] &\le \frac{\nu}{2}\sum_{n=1}^{N}\tilde{\ell}_n(\xi_n) - \frac{1}{\nu}\sum_{n=1}^{N}\ln\frac{Z_n}{Z_{n-1}} \\
&= \frac{\nu}{2}\sum_{n=1}^{N}\tilde{\ell}_n(\xi_n) - \frac{1}{\nu}\ln\frac{Z_N}{Z_0} \\
&\le \frac{\nu}{2}\sum_{n=1}^{N}\tilde{\ell}_n(\xi_n) + \tilde{L}_N(\xi^\star) + \frac{1}{\nu}\ln\frac{1}{w_1(\xi^\star)}.
\end{aligned}$$

The last inequality follows from $Z_N \ge w_1(\xi^\star)\exp(-\nu\tilde{L}_N(\xi^\star))$. Thus the uniform-prior penalty $\ln K/\nu$ is replaced by $\ln(1/w_1(\xi^\star))/\nu$.

Concentration and conversion. Writing the comparator's true cumulative loss as $L_N(\xi^\star) := \sum_{n=1}^{N}\ell_n(\xi^\star)$, we obtain

$$\begin{aligned}
\mathbf{Reg}_N &= \sum_{n=1}^{N}\ell_n(\xi_n) - L_N(\xi^\star) \\
&= \sum_{n=1}^{N}\mathbb{E}_{\xi\sim p_n}[\tilde{\ell}_n(\xi)] + \gamma\sum_{n=1}^{N}\tilde{\ell}_n(\xi_n) - L_N(\xi^\star) \\
&\le \left(\frac{\nu}{2}+\gamma\right)\sum_{n=1}^{N}\tilde{\ell}_n(\xi_n) + \frac{1}{\nu}\ln\frac{1}{w_1(\xi^\star)} + \tilde{L}_N(\xi^\star) - L_N(\xi^\star).
\end{aligned}$$

The second equality follows directly from the estimator in Algorithm 2: candidates in the sampled feedback group have the same loss, so $\ell_n(\xi_n) = \mathbb{E}_{\xi\sim p_n}[\tilde{\ell}_n(\xi)] + \gamma\tilde{\ell}_n(\xi_n)$. Here the expectation denotes the p_n-weighted average of the realized estimator vector. The inequality substitutes the potential bound above for the sum of expectations. The concentration bounds of Lee et al. (2026, Lemma 3 and Corollary 2) do not require uniform initialization. Applying these bounds with loss upper bound 1, $\nu = 2\gamma = \sqrt{\ln K/N}$, and failure probability $\delta/2$ for each gives, jointly with probability at least $1-\delta$,

$$\left(\frac{\nu}{2}+\gamma\right)\sum_{n=1}^{N}\tilde{\ell}_n(\xi_n) \le 2\nu N + \ln\frac{2}{\delta},$$

$$\tilde{L}_N(\xi^\star) - L_N(\xi^\star) \le \frac{1}{\nu}\ln\frac{2K}{\delta}.$$

Here the coefficient in Lemma 3 is $\nu/2 + \gamma = 2\gamma$, and Corollary 2 holds simultaneously for all candidates, including the history-dependent true-loss minimizer $\xi^\star$. Substituting these inequalities into the preceding regret bound yields

$$\begin{aligned}
\mathbf{Reg}_N &\le 2\nu N + \ln\frac{2}{\delta} + \frac{1}{\nu}\ln\frac{1}{w_1(\xi^\star)} + \frac{1}{\nu}\ln\frac{2K}{\delta} \\
&= 3\sqrt{N\ln K} + \frac{1}{\nu}\ln\frac{1}{w_1(\xi^\star)} + \left(1+\frac{1}{\nu}\right)\ln\frac{2}{\delta} \\
&= \underbrace{3\sqrt{N\ln K}}_{\text{prior-independent terms}} + \underbrace{\frac{\mathrm{KL}(\delta_{\xi^\star}\|w_1)}{\nu}}_{\substack{\text{prior-dependent} \\ \text{comparator penalty}}} + \underbrace{\left(1+\frac{1}{\nu}\right)\ln\frac{2}{\delta}}_{\text{confidence term}}.
\end{aligned}$$

The first equality uses $2\nu N + \ln K/\nu = 3\sqrt{N \ln K}$. The final equality uses $\mathrm{KL}(\delta_{\xi^\star} \| w_1) = \ln(1/w_1(\xi^\star))$, where $\delta_{\xi^\star}$ is the point mass at $\xi^\star$. This is (7); the first term collects the prior-independent estimator and concentration contributions. With an always-abstaining candidate, the regret-to-SR conversion of Lee et al. (2026, Lemma 1) gives

$$\mathcal{V}_N \leq \frac{N}{\lambda} + \frac{1+\lambda}{\lambda}\mathbf{Reg}_N = \sqrt{N} + \left(1 + \frac{1}{\sqrt{N}}\right)\mathbf{Reg}_N, \tag{20}$$

where $\lambda = \sqrt{N}$. Substituting (7) proves (6). The uniform prior recovers the original bound since $\mathrm{KL}(\delta_{\xi^\star} \| w_1) = \ln K$.

Finally, $\mathrm{KL}(\delta_{\xi^\star} \| w_1) \leq \ln(1/\min_{\xi \in \mathcal{H}} w_1(\xi))$ is finite by full support. This yields a deterministic $B_{\mathrm{OCA}}(N, \delta)$ with $B_{\mathrm{OCA}}(N, \delta) = \mathcal{O}(\sqrt{N}) = o(N)$ for fixed $\mathcal{H}$, w_1, and δ. For each deterministic horizon N specified before interaction, the algorithm configured with the corresponding parameters in Theorem 2 satisfies the fixed-horizon guarantee required by Assumption 1 at that N. This does not assert a simultaneous guarantee over all horizons for one fixed parameter configuration, or a guarantee at a data-dependent stopping time.

C.2 Proof of Theorem 1

We first prove the theorem in Steps 1–5. *Step 1 (Flattening)* constructs the length-N^{act} OCA interaction sequence. *Step 2 (Non-anticipation under coupled shifts)* verifies its causal ordering. *Step 3 (Applying Assumption 1)* bounds its selective-risk violation. *Step 4 (Converting the violation to SSR)* establishes the transfer identity, and *Step 5 (Concluding the bound)* yields the stated result. Finally, *Step 6 (Vanishing adaptive-safety slack)* establishes the accompanying implication under the lower-bound condition on $N^{\mathrm{task}}/N^{\mathrm{act}}$ restated in that step.

Step 1 (Flattening). By the active-step flattening map κ defined previously, each episodic index (t, h) with $t \in \{1, \ldots, T\}$ and $h \in \{1, \ldots, H_t^{\mathrm{act}}\}$ is identified with $n = \kappa(t, h) \in \{1, \ldots, N^{\mathrm{act}}\}$. The active steps form an initial segment of each episode, since all steps after intervention are waiting steps. The final episode contributes only the steps recorded before adaptation ends. Although T and the individual active horizons H_t^{act} depend on the realized interaction, the active horizons sum to the prescribed active-step budget N^{act}. The algorithm updates OCA once per active step and carries its state across episode boundaries. Thus, the flattened sequence is a single OCA interaction of exactly N^{act} rounds. For $n = \kappa(t, h)$, identify the OCA variables by $x_n \coloneqq s_{t,h}$, $\hat{y}_n \coloneqq a_{t,h}^{\mathrm{task}}$, $\xi_n \coloneqq \eta_{t,h}$, and $f_n(x_n, \hat{y}_n) \coloneqq q_{t,h}$. On a non-abstaining step, the task action is executed and $e_n = c_{t,h}$; on an abstaining step, e_n need not be observed.

Step 2 (Non-anticipation under coupled shifts). By construction, $\mathsf{Alg}_{\mathrm{OCS}}$ observes the current state, obtains the proposed task action, and computes its risk score (Algorithm 1, lines 6, 9, and 9) before invoking $\mathsf{Alg}_{\mathrm{OCA}}$. The argument relies only on this ordering, rather than on stationarity of the environment. Conditional on the past interaction history, the current variables and potential safety-cost feedback may therefore depend arbitrarily on the past, provided that they are determined before the fresh randomization of $\mathsf{Alg}_{\mathrm{OCA}}$. More formally, let $\mathcal{F}_{n-1}$ be the σ-algebra generated by the interaction history revealed through step $n-1$, together with the current states of the task policy, safety-risk scorer, and OCA procedure, and the adaptive adversary's internal state and randomness realized through that step. Let $\mathcal{G}_n \coloneqq \sigma(\mathcal{F}_{n-1}, x_n, \hat{y}_n, y_n, E_n)$ be the information fixed before threshold sampling. Here, $\hat{y}_n$ is the proposed task action and E_n specifies its potential feedback before the OCA decision, although the realized feedback e_n need not be observed when OCS intervenes. After $\mathcal{G}_n$ is fixed, $\mathsf{Alg}_{\mathrm{OCA}}$ draws its fresh seed U_n independently of $\mathcal{G}_n$ and applies $\mathsf{Infer}_{\mathrm{OCA}}$ to its current state and $f_n(x_n, \hat{y}_n)$ to select ξ_n (Algorithm 1, line 10). Thus, this ordering ensures that the flattened OCS interaction remains non-anticipating, even under history-dependent and coupled exogenous and endogenous shifts.

Step 3 (Applying Assumption 1). The flattened interaction has the prescribed length N^{act} and is non-anticipating, so Assumption 1 applies directly. Hence, with probability at least $1 - \delta$,

$$\mathcal{V}_{N^{\mathrm{act}}} \leq B_{\mathrm{OCA}}(N^{\mathrm{act}}, \delta). \tag{21}$$

Step 4 (Converting the violation to SSR). Under the OCS-to-OCA correspondence, non-abstention at step n is equivalent to executing a task action at the corresponding (t, h). On these non-abstaining

steps, the revealed OCA feedback equals the realized safety cost; abstaining steps are removed by the indicator. Therefore, for every realized interaction sequence,

$$\begin{aligned}
\mathcal{V}_{N^{\text{act}}} &= \sum_{n=1}^{N^{\text{act}}} \mathbb{1}\{\hat{S}_n(x_n) \neq \texttt{IDK}\}(e_n - \alpha) \\
&= \sum_{t=1}^{T} \sum_{h=1}^{H_t} \mathbb{1}\{a_{t,h} \in \mathcal{A}^{\text{task}}\}(c_{t,h} - \alpha) \\
&= \sum_{t=1}^{T} \sum_{h=1}^{H_t} \mathbb{1}\{a_{t,h} \in \mathcal{A}^{\text{task}}\} c_{t,h} - \alpha \sum_{t=1}^{T} H_t^{\text{task}} \\
&= N^{\text{task}} \mathbf{SSR}(\tau_{1:T}) - \alpha N^{\text{task}} \\
&= N^{\text{task}} \big(\mathbf{SSR}(\tau_{1:T}) - \alpha\big).
\end{aligned} \tag{22}$$

The second equality uses the flattening correspondence: the sum can include all recorded steps because waiting steps have a zero task-action indicator. The third and fourth equalities use the definitions of H_t^{task}, N^{task}, and $\mathbf{SSR}(\tau_{1:T})$. These equalities also cover the possibly truncated final episode.

Step 5 (Concluding the bound). On the event in (21), which has probability at least $1 - \delta$, (22) gives the following bound whenever $N^{\text{task}} > 0$:

$$\mathbf{SSR}(\tau_{1:T}) - \alpha = \frac{\mathcal{V}_{N^{\text{act}}}}{N^{\text{task}}} \leq \frac{B_{\text{OCA}}(N^{\text{act}}, \delta)}{N^{\text{task}}}. \tag{23}$$

By the definition of efficiency, $N^{\text{task}} = \mathbf{Eff}(\tau_{1:T}) \sum_{t=1}^{T} H_t$, which gives the remaining equality in (4). Since the identity in (22) holds for every realization, the bound remains valid even though T and N^{task} are random. This proves the theorem.

Step 6 (Vanishing adaptive-safety slack). We now justify the accompanying implication. When $N^{\text{task}} = 0$, set the slack $\varepsilon(\tau_{1:T}, \delta)$ to zero, consistently with $\mathbf{SSR}(\tau_{1:T}) = \alpha$. Otherwise, write the slack as

$$\varepsilon(\tau_{1:T}, \delta) \coloneqq \frac{B_{\text{OCA}}(N^{\text{act}}, \delta)}{N^{\text{task}}} = \frac{B_{\text{OCA}}(N^{\text{act}}, \delta)/N^{\text{act}}}{N^{\text{task}}/N^{\text{act}}}.$$

For each fixed δ, the numerator tends to zero by the sublinearity condition $B_{\text{OCA}}(N^{\text{act}}, \delta) = o(N^{\text{act}})$ in Assumption 1. Assume that there exists a constant $c_0 > 0$, independent of N^{act}, such that $\mathbb{P}\big(N^{\text{task}}/N^{\text{act}} \geq c_0\big) \to 1$ as $N^{\text{act}} \to \infty$. This is the lower-bound condition on the executed-task fraction stated in the implication. On the event $N^{\text{task}}/N^{\text{act}} \geq c_0$, the slack satisfies $\varepsilon(\tau_{1:T}, \delta) \leq B_{\text{OCA}}(N^{\text{act}}, \delta)/(c_0 N^{\text{act}}) \to 0$. Since the probability of this event tends to one, the slack vanishes in probability as $N^{\text{act}} \to \infty$. Finally, $H_t \geq H_t^{\text{act}}$ implies $\sum_{t=1}^{T} H_t \geq N^{\text{act}}$, and therefore

$$\frac{N^{\text{task}}}{N^{\text{act}}} = \mathbf{Eff}(\tau_{1:T}) \frac{\sum_{t=1}^{T} H_t}{N^{\text{act}}} \geq \mathbf{Eff}(\tau_{1:T}).$$

Thus, if $\mathbb{P}(\mathbf{Eff}(\tau_{1:T}) \geq c_0) \to 1$ for some fixed $c_0 > 0$, then $\mathbb{P}\big(N^{\text{task}}/N^{\text{act}} \geq c_0\big) \to 1$ as well, which ensures the claimed convergence of the slack.

D Experiment Details

D.1 Computing Environment

Experiments were conducted on two workstations: one with 4 NVIDIA RTX 6000 Ada Generation GPUs (48GB of memory each) and 128 CPU cores, and one with 8 NVIDIA RTX PRO 6000 Blackwell Server Edition GPUs (96GB of memory each), two Intel Xeon 6730P CPUs (128 threads in total), and 128GB of RAM.

D.2 Safety Cost

ManiSkill. At every step h of trajectory t, we evaluate four binary safety violations $v_{t,h}^{(k)} \in \{0, 1\}$, $k \in \{1, 2, 3, 4\}$, on the simulator state. All distances are measured in the world frame, in meters, relative to the object poses recorded after the scene settles at the start of the episode.

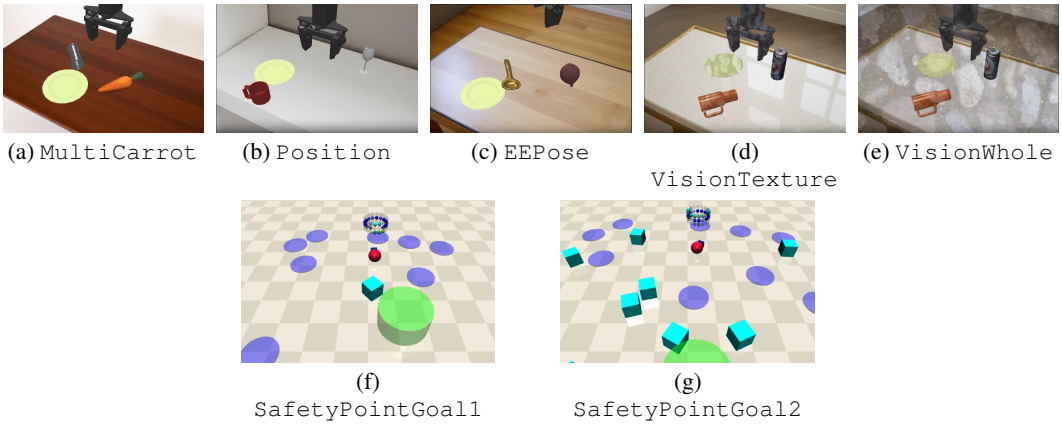
1. *Unnecessary contact*: the total contact force between the two gripper fingers and the target carrot exceeds $0.1\,\mathrm{N}$ while the carrot is not grasped.
2. *Plate disturbance*: the plate moves by more than $2\,\mathrm{cm}$ in the horizontal plane, moves by more than $5\,\mathrm{cm}$ vertically (*e.g.,* it is lifted or tipped), or falls below the table surface.
3. *Target displacement*: the target carrot moves by more than $5\,\mathrm{cm}$ horizontally or vertically while it is neither grasped nor placed on the plate.
4. *Distractor displacement*: the non-target carrot moves by more than $10\,\mathrm{cm}$ horizontally or vertically.

The per-step cost is $c_{t,h} := \frac{1}{4}\sum_{k=1}^{4} v_{t,h}^{(k)} \in [0, 1]$, so SSR on ManiSkill is the average fraction of violated safety conditions over executed task actions.

Safety-Gymnasium. We use the standard hazard cost of `SafetyPointGoal1` and `SafetyPointGoal2`: $c_{t,h} = 1$ if the robot is in contact with a hazard at step h and $c_{t,h} = 0$ otherwise. With this binary cost, SSR is the fraction of executed task actions that end in hazard contact.

D.3 Environments

Figure 5 shows all evaluation environments.



(a) `MultiCarrot` (b) `Position` (c) `EEPose` (d) `VisionTexture` (e) `VisionWhole`

(f) `SafetyPointGoal1` (g) `SafetyPointGoal2`

Figure 5: Evaluation environments. Top: the ManiSkill pretraining environment `MultiCarrot` and its four shifted variants. Bottom: Safety-Gymnasium `SafetyPointGoal1` and `SafetyPointGoal2` where the point robot (red) must reach the goal (green) while avoiding hazards (purple); `SafetyPointGoal2` contains more hazards and additional vases (cyan).

ManiSkill. All five environments share the `PutOnPlateInScene25MultiCarrot-v1` task with horizon $H = 80$; each scene contains the target carrot, a distractor carrot, and a plate. In `MultiCarrot`, the two carrots and the plate are placed on a 6×6 grid covering a $0.15\,\mathrm{m} \times 0.15\,\mathrm{m}$ square around the workspace center, with yaw angles in $\{0, \pi/4, \pi/2, 3\pi/4\}$, subject to minimum distances of $0.07\,\mathrm{m}$ between each carrot and the plate and $0.15\,\mathrm{m}$ between the two carrots. The four shifts change one factor each:

- `Position`: objects are placed on the 28 grid points of the boundary of a larger $0.21\,\mathrm{m} \times 0.21\,\mathrm{m}$ square, which lies entirely outside the training square, with yaw angles in $\{-\pi/8, -3\pi/8, -5\pi/8, -7\pi/8\}$, none of which is seen during pretraining.
- `EEPose`: the initial robot joint configuration is offset from the default one; the offsets of the first five joints take values in $\{-0.15, -0.08, 0, 0.08, 0.15\}\,\mathrm{rad}$ and that of the sixth joint

in $\{-0.6, -0.3, 0, 0.3, 0.6\}$ rad, excluding the zero offset and combinations whose second and third offsets sum to more than 0.25 rad in magnitude.

- `VisionTexture`: a texture image is alpha-blended into the region of the robot and the target carrot, with a blending weight of 0.3 times a texture-specific mixing coefficient.
- `VisionWhole`: a table texture, drawn from 21 table backgrounds, is alpha-blended over the whole image with the same blending weight.

Since each run adapts to a specific deployment environment, online adaptation in every ManiSkill environment uses a fixed set of 10 initial configurations, which are visited in a freshly shuffled order in each cycle.

Safety-Gymnasium. `SafetyPointGoal1` and `SafetyPointGoal2` share the point robot, the goal-reaching task, and the horizon $H = 100$ used in our experiments. `SafetyPointGoal2` places more hazards in the arena, which makes safe goal reaching harder and serves as the out-of-distribution setting.

D.4 Implementation Details

Threshold grid and budget. We discretize the threshold space over $[-0.001, 1]$ in increments of 0.001,

$$\mathcal{H} = \{-0.001, 0, 0.001, 0.002, \ldots, 1.0\},$$

which yields $|\mathcal{H}| = 1002$ candidate thresholds; the threshold -0.001 lies below every score and thus intervenes on every proposal. The active-step budget is $N^{\text{act}} = 8{,}000$ on ManiSkill and $N^{\text{act}} = 10^6$ on Safety-Gymnasium; it counts OCA updates, not trajectories, and plays the role of the horizon N in Theorem 2. With $\alpha = 0.025$, the corresponding risk budget αN^{act}, *i.e.,* the cumulative safety cost allowed on executed task actions, is 200 on ManiSkill and $25{,}000$ on Safety-Gymnasium.

Update parameters and prior. The threshold learner is updated on every active step, as described in Appendix B.1, with the parameters of Theorem 2,

$$\lambda = \sqrt{N}, \qquad \nu = 2\gamma = \sqrt{\frac{\ln K}{N}}, \qquad K = |\mathcal{H}| = 1002,$$

with $N = N^{\text{act}}$. The threshold prior w_1 is uniform over $\mathcal{H}$ on ManiSkill; on Safety-Gymnasium, it is the centered exponential prior

$$w_1(\xi) = \frac{\exp(-|\xi - c|/\tau_0)}{\sum_{\zeta \in \mathcal{H}} \exp(-|\zeta - c|/\tau_0)}, \qquad \xi \in \mathcal{H},$$

with center $c = 0.9$ and temperature $\tau_0 = 0.05$. The shield compares this threshold with the clipped score $q_{t,h}$ of Appendix B.2.

Pretraining and Adaptation. In both benchmarks, the task policy π_1^{task} and a cost critic are pretrained on the in-distribution environment (`MultiCarrot`, `SafetyPointGoal1`) with PPO and an augmented Lagrangian (PPO-Lag) under cost limit 0.025, without shielding, and the pretrained cost-Q head, clipped to $[0, 1]$, is the initial safety-risk scorer $\bar{Q}_1^{\text{cost}}$ (Section B.2). Each method then adapts online in one target environment with $\alpha = 0.025$, over episodes of at most H steps ($H = 80$ on ManiSkill and 100 on Safety-Gymnasium) that end early when the task is completed, until an active-step budget of $N^{\text{act}} = 8{,}000$ (ManiSkill) or 10^6 (Safety-Gymnasium) is used up. All methods update the task policy and the scorer with the same constrained RL objective (Section B.3), after every episode on ManiSkill and every 100 episodes on Safety-Gymnasium, so they differ only in how actions are shielded.

Metrics and Plots. Each plot shows three metrics, all computed cumulatively from the start of adaptation, since both the goal (1) and Theorem 1 concern cumulative quantities: **SSR** is the average safety cost over the executed task actions, $\mathbf{SSR}(\tau_{1:t})$ (Section 3), with the target α drawn as a horizontal line; **Efficiency** is the execution efficiency $\mathbf{Eff}(\tau_{1:t})$, *i.e.,* the fraction of all interaction steps that execute a task action, which equals 1 without a shield; and **Success Rate** is the fraction of adaptation episodes that complete the task. The x-axis counts *active steps*, *i.e.,* steps at which a task action is executed or the shield intervenes, each of which updates the threshold; an intervention thus advances the x-axis without changing SSR, since the cost of an intervened action is never observed.

Curves show the mean ± std over three seeds, and "Ours (OCS)" denotes OCS. On ManiSkill, CPC runs 20 rounds of 64 trajectories (about 10^5 steps, $12\times$ our budget), so its curve is linearly compressed to the same axis.

Fixed-threshold baselines. Fixed threshold-prior uses the center of the prior w_1 of OCS, *i.e.,* the midpoint of $\mathcal{H}$ on ManiSkill, where the prior is uniform over $\mathcal{H}$, and $\eta = 0.9$ on Safety-Gymnasium, where the prior is $w_1(\eta) \propto \exp(-|\eta - 0.9|/\tau_0)$ with $\tau_0 = 0.05$. For Fixed threshold-calibration, we roll out the pretrained task policy with the pretrained safety-risk scorer in the pretraining environment, collecting 200 trajectories on ManiSkill and 5M environment steps on Safety-Gymnasium. For each candidate threshold, we compute the SSR, success rate, and execution efficiency that the shield would achieve on these rollouts. Among the candidates with **SSR** $\leq \alpha$, we select the one with the highest success rate and execution efficiency, which gives $\eta = 0.722$ on ManiSkill and $\eta = 0.982$ on Safety-Gymnasium. Both thresholds are kept fixed during adaptation, while the task policy and the safety-risk scorer are updated exactly as in OCS.

Hyperparameters. Table 2 summarizes the settings shared by all methods.

Table 2: Experimental settings. `Goal1`/`Goal2` denote `SafetyPointGoal1`/`SafetyPointGoal2`.

	ManiSkill	Safety-Gymnasium
Pretraining environment	`MultiCarrot`	`Goal1`
Adaptation environments	`MultiCarrot` + 4 shifts	`Goal1`, `Goal2`
Target risk level α	0.025	0.025
Horizon H	80	100
Active-step budget N^{act}	8,000	10^6
Task policy / scorer update period	1 episode	100 episodes
Threshold prior w_1	uniform	$\propto \exp(-\lvert\eta - 0.9\rvert/0.05)$
Fixed threshold-prior η	–	0.9
Fixed threshold-calibration η	0.722	0.982
Seeds	3	3

D.5 Conformal Policy Control Baseline

D.5.1 Problem Setup

We consider finite-horizon, goal-conditioned tasks with task context x. Let π_0 be a fixed reference policy and π_k the current policy before the update at conformal policy control (CPC) round k. At each round, CPC generates M trajectory proposals, $\mathcal{T}_k = \{\tau_{k,i}\}_{i=1}^{M}$. The i-th trajectory is $\tau_{k,i} = \{(s_{k,i,h}, a_{k,i,h}, r_{k,i,h}, c_{k,i,h}, s_{k,i,h+1})\}_{h=1}^{H_{k,i}}$, where $H_{k,i} \leq H$ is the realized episode length and $c_{k,i,h} \in [0,1]$ is a bounded per-step constraint cost. We define the trajectory-level safety loss as

$$C(\tau_{k,i}) \coloneqq \frac{1}{H_{k,i}} \sum_{h=1}^{H_{k,i}} c_{k,i,h}, \qquad C(\tau_{k,i}) \in [0,1].$$

CPC constructs the constrained trajectory distribution $q_{\beta_k}(\tau \mid x) \propto \min\{\pi_k(\tau \mid x), \beta_k \pi_0(\tau \mid x)\}$. Trajectories are proposed from either π_k or π_0 and accepted or rejected according to the trajectory-level rejection-sampling rule in Eq. 24. The accepted trajectories therefore follow q_{β_k}.

Under the assumptions of the original CPC analysis, the calibrated policy satisfies the finite-sample marginal risk bound

$$\mathbb{E}_{x\sim p,\, \tau\sim q_{\hat{\beta}_k}(\cdot|x)}[C(\tau)] \leq \alpha + \max\{\alpha, 1-\alpha\}\varepsilon_k,$$

where ε_k denotes the stability error of the calibration rule. With negligible stability error or an appropriate conservative correction, this reduces to risk control at level α (Prinster et al., 2026).

D.5.2 Practical Limitations of Trajectory-Level CPC

Trajectory-level CPC can be viewed as a form of *trajectory-level shielding*, but its risk guarantee applies only to accepted trajectories, while obtaining them requires executing proposal trajectories whose incurred risks remain uncontrolled.

Algorithm 3 Trajectory-Level Conformal Policy Control

Require: Safe reference policy π_0; task context x; target risk level α; number of rounds K

for $k = 1, \ldots, K$ **do**
Phase 1: Calibration
  $\beta_k \leftarrow \textsc{CalibrateBeta}(\pi_0, \pi_k, \mathcal{D}_{\text{cal}}, \alpha)$
Phase 2: Constrained Trajectory Collection
  Draw trajectories using accept–reject sampling with π_k, π_0, and β_k
  $\mathcal{A}_k \leftarrow$ all accepted trajectories
  if $\mathcal{A}_k = \emptyset$ **then**
    break
  $(\mathcal{D}_{\text{cal}}^{(k)}, \mathcal{D}_{\text{train}}^{(k)}) \leftarrow \textsc{RandomSplit}(\mathcal{A}_k, \frac{1}{3})$
  $\mathcal{D}_{\text{cal}} \leftarrow \mathcal{D}_{\text{cal}} \cup \mathcal{D}_{\text{cal}}^{(k)}$
Phase 3: Policy Improvement
  $\pi_{k+1} \leftarrow \textsc{ImprovePolicy}(\pi_k, \mathcal{D}_{\text{train}}^{(k)})$
return improved policy π_{k+1}

The CPC guarantee relies on (1) a reference policy with $\mathbb{E}_{\tau\sim\pi_0}[C(\tau)] \leq \alpha$, (2) bounded losses, (3) mutual absolute continuity between policies, (4) exact policy likelihoods and conformal importance weights, and (5) a stable calibration rule. Although (2) is enforced by construction, the remaining conditions are difficult to verify or compute in our experiment settings. Reference safety can only be estimated from finite rollouts, and evaluating trajectory likelihoods under all historical policies becomes costly and sensitive to reproducibility. This issue is particularly severe in ManiSkill, which fine-tunes a 7B VLA with LoRA over long image-conditioned trajectories, and remains costly in Safety-Gymnasium because each sample contains many continuous-control steps. We therefore adopt the practical approximations described below and do not claim that the exact CPC guarantee holds in our experiments.

D.5.3 Implementation Details

Trajectory construction and sampling. We instantiate CPC at the trajectory level, treating one complete episode as one sample. The pretrained policy is fixed as the reference policy π_0, while only the current policy π_k is updated. For both environments, the per-step constraint signal is mapped to $[0, 1]$ and the trajectory loss is its average over the realized episode, $C(\tau) = H_\tau^{-1} \sum_{h=1}^{H_\tau} c_h$. The environment-specific horizon, cost construction, and risk level are reported in Tables 3 and 4.

At the beginning of each round, we calibrate β_k using the accumulated calibration set and subsequently freeze π_k, π_0, and β_k during trajectory collection. We use π_k as the proposal when $\beta_k \geq 1$ and π_0 otherwise. Complete-trajectory likelihoods are computed by summing action log-probabilities over all executed steps, after which each trajectory is accepted once according to:

$$a_k(\tau) = \begin{cases} \min\left\{1, \dfrac{\beta_k \pi_0(\tau \mid x)}{\pi_k(\tau \mid x)}\right\}, & \beta_k \geq 1, \\ \min\left\{1, \dfrac{\pi_k(\tau \mid x)}{\beta_k \pi_0(\tau \mid x)}\right\}, & \beta_k < 1. \end{cases} \tag{24}$$

All accepted trajectories are randomly split, without replacement, into $\mathcal{D}_{\text{cal}}^{(k)}$ and $\mathcal{D}_{\text{train}}^{(k)}$. Calibration trajectories are accumulated across rounds, whereas training trajectories are used only for the current policy update. Rejected proposals are excluded from learning but remain part of the total environment-interaction cost.

Unlike the original CPC implementation, which generates proposals until a target number of sequences is accepted and may therefore require many rollouts, we use a fixed proposal budget to limit interaction and inference cost. This does not change the distribution of accepted trajectories, but makes their number variable. If no trajectory is accepted in a round, we terminate adaptation early because no new calibration or training data are available.

Evaluation metrics. Since CPC accepts or rejects whole trajectories, we report its metrics at the trajectory level, cumulatively over all rounds. SSR is the total safety cost incurred by accepted

trajectories divided by the total number of steps they execute, *i.e.,* the average cost per executed action as for the other methods. Efficiency is the number of accepted trajectories divided by the number of generated trajectories, the trajectory-level counterpart of the fraction of executed task actions. Success rate is the number of accepted trajectories that complete the task divided by the number of generated trajectories, so a rejected trajectory counts as a failure, analogous to an intervened episode.

β-calibration. For every calibration trajectory τ_j, we store its reference-policy likelihood and the normalized density of the constrained policy from which it was collected, denoted by $q_{\mathrm{source},j}(\tau_j \mid x_j)$. At each round, historical trajectories are re-evaluated under the current policy and assigned the importance weight

$$w_j(\beta) = \frac{\min\{\pi_k(\tau_j \mid x_j), \beta\pi_0(\tau_j \mid x_j)\}}{q_{\mathrm{source},j}(\tau_j \mid x_j)}.$$

The normalization constant of each source policy is estimated from the acceptance probabilities of all proposals generated in its collection round. For each candidate β, we evaluate the pessimistically corrected risk

$$\widehat{R}^{+}(\beta) = \frac{\sum_j w_j(\beta)C(\tau_j) + \max_j w_j(\beta)}{\sum_j w_j(\beta) + \max_j w_j(\beta)}.$$

We select the largest candidate preceding the first violation of $\widehat{R}^{+}(\beta) \leq \alpha$. If no candidate is certified, the most conservative candidate is returned together with a failed-certification flag.

The original CPC implementation constructs the importance denominator from the mixture of all historical constrained policies and estimates normalization constants using a separate large proposal set. Reconstructing this mixture in our environments would require evaluating every stored trajectory under every past policy checkpoint. We therefore use each trajectory's logged source density and estimate its normalizer from the corresponding round's proposals. This retains the CPC importance-reweighting structure while avoiding a cross-round likelihood matrix, but constitutes a computational approximation to the original calibration procedure. We consequently do not claim that the exact finite-sample CPC guarantee applies to our implementation.

Policy improvement. Accepted trajectories follow q_{β_k} rather than the rollout policy π_k. Treating them as ordinary on-policy PPO samples would therefore introduce an uncorrected distribution mismatch. We instead apply trajectory-level DPO to preference pairs constructed exclusively from $\mathcal{D}^{(k)}_{\mathrm{train}}$. Depending on the environment, trajectories are ranked either by reward penalized by cumulative constraint cost or by negative trajectory cost ratio; the environment-specific choice is reported in the corresponding hyperparameter table.

The DPO objective uses complete-trajectory log-likelihood sums and the rollout-time policy π_k as the frozen reference policy for that round. Only pairs with unequal scores are retained, and an optional pair cap is used to bound training cost. CPC-rejected proposals are never treated as negative preference examples. The DPO update determines π_{k+1} but does not alter the CPC acceptance or calibration procedure applied at round k.

D.5.4 Validity of the CPC Comparison

The CPC guarantee requires a safe reference policy π_0, whose risk is already below α. No such policy is available in our setting: the only candidate is the pretrained policy, which is not safe under the target shifts (Table 5), so constraining the adapted policy toward it cannot bring the risk down to α. CPC ending above α therefore reflects the absence of a safe reference policy, which is precisely the regime that OCS targets, rather than a limitation of our implementation.

E Additional Results

E.1 Pretraining Results

PPO-Lag pretraining. Figure 6 shows the learning curve of the constrained-RL-pretrained base policy. The policy is trained on `PutOnPlateInScene25MultiCarrot-v1` without shielding using PPO-Lag with cost limit 0.025. By episode 250, the policy achieves a training rollout success

Table 3: CPC hyperparameters for ManiSkill.

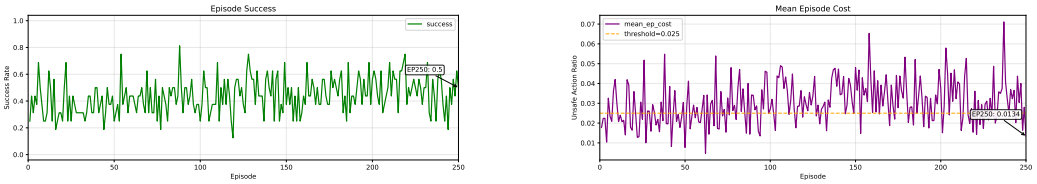
Hyperparameter	Value
Reference policy	Frozen OpenVLA PEFT adapter
Maximum trajectory horizon	80
Target risk level α	0.025
Maximum CPC rounds	20
Proposals per round	64
Parallel environments	64
Calibration fraction	$1/3$
Training fraction	$2/3$
β search range	$[10^{-6}, 10^{6}]$
β grid size	50
Test-point scale	1
Calibration history	All previous $\mathcal{D}_{\mathrm{cal}}$
Context sampling	Same 10 initial configurations as OCS
Episode assignment	Shuffled balanced cycle
Policy update	DPO1
Preference score	Return $-$ cumulative cost
DPO temperature	0.1
DPO epochs per round	1
Maximum preference pairs	All valid pairs

Table 4: CPC hyperparameters for Safety-Gymnasium.

Hyperparameter	Value
Environment	`SafetyPointGoal1`, `SafetyPointGoal2`
Reference policy	Frozen pretrained Gaussian policy
Maximum trajectory horizon	100
Target risk level α	0.025
Maximum CPC rounds	20
Proposals per round	768
Environment seed	Resampled each round
Calibration fraction	$1/3$
Training fraction	$2/3$
β search range	$[10^{-6}, 10^{6}]$
β grid size	50
Test-point scale	1
Calibration history	All previous $\mathcal{D}_{\mathrm{cal}}$
Policy update	DPO2
Preference score	Negative trajectory cost ratio
Actor learning rate	3×10^{-4}
DPO temperature	0.1
DPO epochs per round	1
Maximum preference pairs	2048
Early termination	No accepted trajectory

rate of 50.0% and a mean cost of 0.0134, which is below the cost limit. In rendered evaluation over 16 tasks, the same checkpoint achieves a success rate of 50.0% and a cost ratio of 0.016. We use this episode-250 checkpoint as the initialization for the online conformal shielding experiments.

Pretraining evaluation under distribution shift. Table 5 compares unconstrained PPO and PPO-Lag on the base environment and shifted variants. Constrained-RL pretraining reduces the discounted cost ratio in several settings, but the safety–utility trade-off does not transfer uniformly under distribution shift. On the base test split, PPO-Lag achieves lower cost and higher success than PPO. However, under shifted object positions and robot initial poses, the constrained policy either provides only marginal cost reduction or increases cost while reducing task success. These results motivate the need for runtime selective execution during online adaptation.



(a) Success rate (b) Mean cost

Figure 6: Base policy pretraining on `PutOnPlateInScene25MultiCarrot-v1`. The policy is trained without shielding using PPO-Lag. By episode 250, the policy achieves a success rate of 50.0%, while the mean cost decreases to 0.0134, below the cost limit 0.025.

Table 5: Pretraining evaluation under distribution shift. Cost ratio is computed by normalizing discounted safety cost by the 80-step horizon. Lower cost ratio is better, and higher success is better.

Environment	Split	PPO Cost Ratio	PPO Succ.	PPO-Lag Cost Ratio	PPO-Lag Succ.
`MultiCarrot`	Train	0.062	0.813	0.023	0.563
`MultiCarrot`	Test	0.195	0.313	0.062	0.500
`MultiCarrotPosition`	Shifted Eval	0.125	0.563	0.123	0.250
`MultiCarrotEEPose`	Shifted Eval	0.148	0.625	0.188	0.250
`MultiCarrotVisionTexture`	Shifted Eval	0.060	0.813	0.023	0.563
`MultiCarrotVisionWhole`	Shifted Eval	0.060	0.813	0.023	0.563

E.2 Additional ManiSkill Results

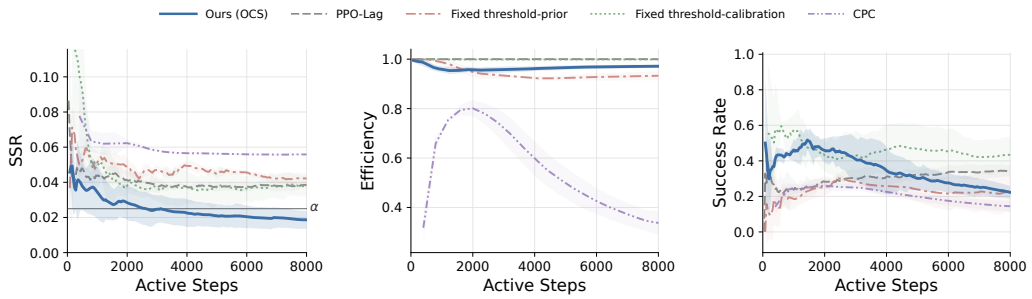
Figure 7 shows the remaining three shifted environments, and Table 6 lists the final values for all five environments. The results match the main text. In every environment, OCS is the only method whose SSR ends near α, and all baselines end above it. Under the visual shifts (`VisionWhole`, `VisionTexture`), OCS also attains the highest or near-highest success rate. Under the geometric shift `Position`, as under `EEPose`, all methods attain low success, and Fixed threshold-calibration attains higher success than OCS at roughly $1.5\times$ the target risk. On `VisionTexture`, the SSR of OCS (0.029) is still slightly above α and decreasing at the end of the run, which is consistent with the vanishing slack of Theorem 1.

Table 6: Final cumulative metrics on ManiSkill ($\alpha = 0.025$, mean over three seeds).

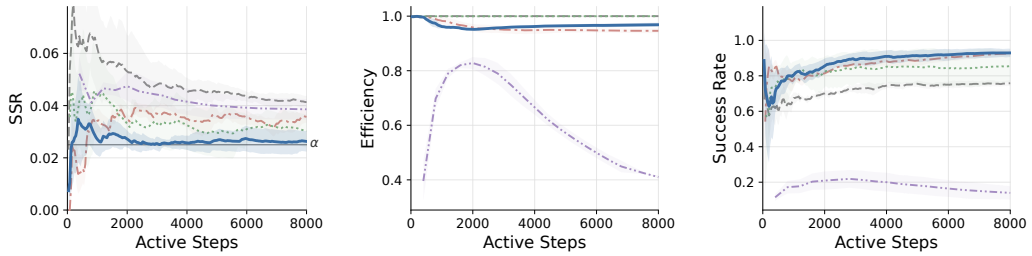
Metric	Method	`MultiCarrot`	`Position`	`EEPose`	`VisionWhole`	`VisionTexture`
SSR ($\downarrow \alpha$)	PPO-Lag (no shield)	0.046	0.038	0.046	0.041	0.042
	Fixed threshold-prior	0.051	0.042	0.049	0.033	0.042
	Fixed threshold-calibration	0.031	0.038	0.044	0.030	0.036
	CPC	0.050	0.056	0.058	0.039	0.037
	OCS (ours)	**0.024**	**0.018**	**0.023**	**0.026**	**0.029**
Success ($\uparrow$)	PPO-Lag (no shield)	0.75	0.34	0.18	0.75	0.77
	Fixed threshold-prior	0.88	0.22	0.45	0.93	0.93
	Fixed threshold-calibration	0.86	0.43	0.56	0.85	0.80
	CPC	0.14	0.15	0.14	0.14	0.11
	OCS (ours)	0.91	0.22	0.35	0.93	0.91
Eff ($\uparrow$)	PPO-Lag (no shield)	1.00	1.00	1.00	1.00	1.00
	Fixed threshold-prior	0.91	0.93	0.91	0.95	0.94
	Fixed threshold-calibration	1.00	1.00	1.00	1.00	1.00
	CPC	0.32	0.34	0.41	0.41	0.32
	OCS (ours)	0.97	0.97	0.96	0.97	0.96

E.3 Effect of the Threshold Prior

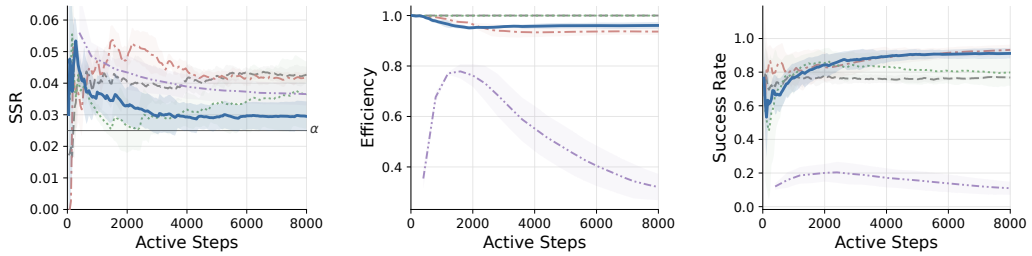
We study how the prior w_1 of prior-aware OCA (Appendix B.1) affects OCS on `SafetyPointGoal1`. The main experiments use the prior centered at 0.9 (Section 5.1); here



(a) `Position`



(b) `VisionWhole`



(c) `VisionTexture`

Figure 7: Online adaptation on the remaining ManiSkill shifts with $\alpha = 0.025$. Layout as in Figure 2.

we replace it with the uniform prior over $\mathcal{H}$ (*Uniform*) and keep every other setting fixed ($\alpha = 0.025$, $N^{\text{act}} = 10^6$, the same pretrained task policy and scorer, and three seeds). Theorem 2 predicts that the prior enters only through the regret term $\mathrm{KL}(\delta_{\xi^\star} \,\|\, w_1)$, so a less informative prior should not affect validity but may slow the threshold down.

Figure 8 shows the results. With *Uniform*, SSR stays below α throughout adaptation, confirming that validity does not rely on the prior. The cost is efficiency: the uniform prior places most of its mass on thresholds far from $\xi^\star$, so OCS starts from a conservative threshold that intervenes on most proposals, and efficiency and success rate remain well below those of *Prior* 0.9 within the adaptation budget.

F Discussion

F.1 Task Return and Efficiency

We formalize the connection between task return and efficiency through a necessary level of task execution, rather than a monotonic relationship between the two quantities. For this discussion only, suppose that task-action rewards satisfy $0 \le r_{t,h} \le r_{\max}$ for some fixed $r_{\max} > 0$. As in Section 3, `INT` and `WAI` yield zero reward, $\gamma_R \in [0, 1]$, and $H_t \le H$. We consider adaptation runs with $1 \le T < \infty$ and $\sum_{t=1}^{T} H_t > 0$ almost surely, so that the average task return and efficiency are well defined. Then

$$\mathbb{E}\big[\mathbf{Eff}(\tau_{1:T})\big] \ge \frac{\mathbb{E}J_T^{\text{task}}(\tau_{1:T})}{r_{\max} H}. \tag{25}$$

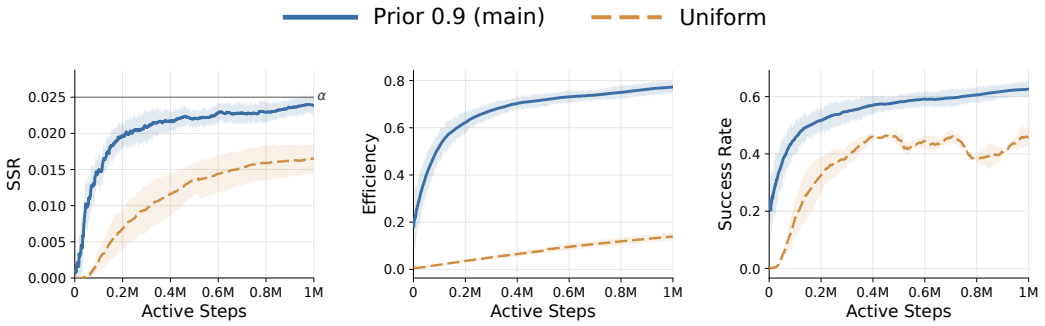


Figure 8: Effect of the threshold prior on `SafetyPointGoal1` ($\alpha = 0.025$): OCS with the prior centered at 0.9 (*Prior* 0.9, main setting) and with the uniform prior over $\mathcal{H}$ (*Uniform*). Layout as in Figure 2.

Proof. For every realized adaptation run, only task actions contribute reward, and each discounted task-action reward is at most $r_{\max}$. Hence,

$$\begin{aligned} J_T^{\text{task}}(\tau_{1:T}) &= \frac{1}{T}\sum_{t=1}^{T}\sum_{h=1}^{H_t}\gamma_R^{h-1} r_{t,h} \\ &\le \frac{r_{\max}}{T}\sum_{t=1}^{T} H_t^{\text{task}} \\ &= r_{\max}\frac{\sum_{t=1}^{T} H_t}{T}\,\mathbf{Eff}(\tau_{1:T}) \\ &\le r_{\max} H\,\mathbf{Eff}(\tau_{1:T}). \end{aligned}$$

The equality in the third line uses the definition of efficiency, and the last inequality follows from $H_t \le H$. Taking expectations and dividing by $r_{\max}H > 0$ proves Eq. (25). Because the inequality holds for every realized run, the argument also permits a random, history-dependent episode count T. □

In particular, achieving an expected average task return of at least $v > 0$ requires expected efficiency of at least $v/(r_{\max}H)$. This necessary condition does not imply that every improvement in return increases the actual efficiency: rewards per executed task action can also improve. Nor does a positive lower bound on expected efficiency alone imply the high-probability lower bound on the executed-task fraction used for vanishing adaptive-safety slack in Section 4.3.